

# ແອປລີເຄັ້ນຕົວໜ່ວຍໃນກາຣດູແລແພດກດທັບ

## DECUBITUS

ກວັດ ແຢ້ມວງ

ຕະວິ່ຈຸ່ງ ເລຍວານິ່ຈິ່ງ

ນັ້ນວາ ສັກດາຮັດນໍ

ສຶກສີ ນະນະວັດນໍ

ປະລຸງລູນິພນ໌ນີ້ເປັນສ່ວນໜຶ່ງຂອງກາຣສຶກໝາຕາມຫລັກສູດປະລຸງລູນິວິສະວົງກຣມຄາສຕຣບັນທຶກ  
ສາຂາວິຊາວິສະວົງກຣມຄອມພິວເຕອ້ນ ຄະວິສະວົງກຣມຄາສຕຣ  
ສາທັນເກໂຄໂນໂລຢີພະຈອມເກົ່າເຈົ້າຄູນທຫາຣລາດກະບັງ  
ປີກາຣສຶກໝາ 2567

ปริญญาบัตรชั้นปีการศึกษา 2567

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แอปพลิเคชันตัวช่วยในการดูแลแพลกัดทับ

DECUBITUS

ผู้จัดทำ

1. กวัต แย้มวงศ์ รหัสนักศึกษา 64010646
2. สรวิชญ์ เลยวนิชย์เจริญ รหัสนักศึกษา 64010876
3. มัณฑ์ ศักดาธัตน์ รหัสนักศึกษา 64011148
4. สิทธิ นวะมะวัฒน์ รหัสนักศึกษา 64011301

อรอนงค์ ทิรุ

อาจารย์ที่ปรึกษา

(รศ.ดร.อรอนงค์ จิตต์ไสภักษตร์)

## แอปพลิเคชันตัวช่วยในการดูแลผู้ป่วยทับ

นายกวัต	แย้มวงศ์	64010646
นายสุริชญ์	เดลิวนิชญ์เจริญ	64010876
นายธนวัฒน์	ศักดาธนวัฒน์	64011148
นายสิทธิ์	นพะมะวัฒน์	64011301
รศ.ดร.อรุณตระ	จิตต์โสภกัตร์	อาจารย์ที่ปรึกษา
ปีการศึกษา 2567		

### บทคัดย่อ

แอปพลิเคชัน DECUBITUS เป็นแอปพลิเคชันที่รวมความสามารถของ Telemedicine App ระบบจัดการผู้ป่วยในโรงพยาบาล และตัวช่วยในการทำงานความรุนแรงของแพลงก์ทับ เพื่อใช้ในการลดภาระงานของบุคลากรทางการแพทย์ ตามความต้องการทางการแพทย์ที่เพิ่มมากขึ้นในปัจจุบัน โดยแอปพลิเคชัน DECUBITUS จะช่วยให้ผู้ใช้งานสามารถจำแนกแพลงก์ทับในเบื้องต้นได้ ได้เรียนรู้วิธีการดูแลรักษาแพลงก์ทับเบื้องต้น และเป็นตัวช่วยในการหาแนวทางการบรรเทาอาการของแพลงก์ทับผู้ป่วยได้ รวมถึงผู้ป่วยยังสามารถได้รับการติดตามอาการของแพลงก์ทับแพทย์และพยาบาลได้ สุดท้ายแอปพลิเคชันนี้ยังเป็นเพียงตัวช่วยในการตัดสินใจของแพทย์และพยาบาลเพื่อใช้ในการดูแลคนไข้ที่มีแพลงก์ทับ

# **DECUBITUS**

Mr. Pawat	Yamwong	64010646
Mr. Sorawit	Loeyvanicharoen	64010876
Mr. Thanwa	Sakdarat	64011148
Mr. Sitthi	Nawamawat	64011301
Assoc.Prof.Dr. Orachat	Chitsobhuk	Advisor
Academic Year 2024		

## **ABSTRACT**

The DECUBITUS application integrates the capabilities of a telemedicine app, hospital patient management system, and a tool for predicting the severity of pressure ulcers. It aims to alleviate the workload of medical personnel, given the growing demand for healthcare services. DECUBITUS allows users to initially classify pressure ulcers, learn basic wound care methods, and explore ways to alleviate symptoms for patients. Additionally, patients can receive ongoing wound monitoring from doctors and nurses. Finally, the application serves as a decision-support tool for healthcare providers in managing and treating patients with pressure ulcers.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณาและความอนุเคราะห์จากอาจารย์ที่ปรึกษา  
รศ.ดร.อรณัตร จิตต์โสภกตร์ ที่ให้คำปรึกษา คำแนะนำ รวมถึงการชี้แนะต่างๆ ตลอดจนให้ความรู้ที่  
ดีแก่สมาชิกกลุ่มปริญญานิพนธ์

ขอกราบขอบพระคุณครองครัวของกลุ่มปริญญานิพนธ์นี้ ที่การอุปการะอบรมเลี้ยงคู่  
รวมถึงให้คำปรึกษาแนะนำต่างๆ ตลอดจนส่งเสริมให้โอกาสทางการศึกษา และให้กำลังใจกลุ่ม  
ปริญญานิพนธ์นี้เป็นอย่างดี

สำหรับประโยชน์อันได้เกิดจากปริญญานิพนธ์ฉบับนี้ พากเราขอบมอบให้กับบุคลากรค่า  
บูรพาจารย์ รวมทั้งผู้มีพระคุณทุกท่านที่รักและเคารพยิ่ง

กวัต แม้มวงศ์

สรวิชญ์ เลยานิชย์เจริญ

ธันวา ศักดิรัตน์

สิทธิ นวมมะวัฒน์

# สารบัญ

	หน้า
แอปพลิเคชันตัวช่วยในการดูแลแพลกอดทับ.....	I
DECUBITUS .....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญตาราง .....	VI
สารบัญรูป .....	VIII
 บทที่ 1 บทนำ .....	 1
1.1 ความเป็นมาของปัญหา .....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ .....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ .....	2
 บทที่ 2 ทฤษฎีและเครื่องมือที่เกี่ยวข้อง .....	 3
2.1 เครื่องมือที่ใช้ในโครงการ .....	3
2.2 ทฤษฎีที่ใช้งานวิจัย .....	15
2.3 งานวิจัยที่เกี่ยวข้อง .....	25
2.4 แอปพลิเคชันที่เกี่ยวข้อง .....	31
 บทที่ 3 การออกแบบ .....	 33
3.1 ภาพรวมของระบบ .....	33
3.2 การออกแบบสถาปัตยกรรมระบบซอฟต์แวร์ .....	36
3.3 การออกแบบ USE CASE DIAGRAM .....	37
3.4 การออกแบบ SEQUENCE DIAGRAM .....	54
3.5 การออกแบบฐานข้อมูล.....	62
3.6 การออกแบบ UI .....	69

## สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการดำเนินงาน .....	76
4.1 การพัฒนาและออกแบบในส่วนแอปพลิเคชัน .....	76
4.3 การพัฒนาระบบสำหรับผู้ดูแลระบบและเดชบอร์ด .....	81
4.3 การพัฒนาระบบในส่วน API .....	86
4.4 การพัฒนาระบบในส่วนของโมเดล .....	95
4.5 การติดตั้งใช้งาน (DEPLOYMENT) ระบบในภาพรวม .....	113
บทที่ 5 สรุปผลการดำเนินงาน .....	115
5.1 สรุปผลการดำเนินงาน .....	115
5.2 ปัญหาและแนวทางแก้ไข .....	116
5.3 แนวทางการพัฒนาต่อ .....	116
บรรณานุกรม .....	117

# สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.2 ตารางเปรียบเทียบเมทริกซ์ของผลลัพธ์ที่ได้จากโมเดลแต่ละตัวในงานวิจัย FULLY AUTOMATIC WOUND SEGMENTATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS .....	27
ตารางที่ 2.3 ตารางเปรียบเทียบปริมาณรูปบาดแผลแต่ละประเภท .....	29
ตารางที่ 2.4 ตารางเปรียบเทียบเมทริกซ์ของผลลัพธ์ที่ได้จากโมเดลแต่ละตัวในงานวิจัย DEEP LEARNING FOR AUTOMATED WOUND CLASSIFICATION AND SEGMENTATION .....	31
ตารางที่ 2.1 ตารางเปรียบเทียบ FUNCTION ของแอปพลิเคชัน IMITOMEASURE .....	32
ตารางที่ 3.1 REGISTER USE CASE .....	39
ตารางที่ 3.2 LOGIN USE CASE .....	40
ตารางที่ 3.3 REVIEW WOUND RECORDS INFORMATION USE CASE .....	41
ตารางที่ 3.4 FOLLOW UP WOUND RECORDS INFORMATION USE CASE .....	42
ตารางที่ 3.5 CHAT (USER) USE CASE .....	43
ตารางที่ 3.6 CHAT (NURSE) USE CASE .....	43
ตารางที่ 3.7 APPOINTMENT VIDEO CALL USE CASE .....	44
ตารางที่ 3.8 REVIEW WOUND RECORDS INFORMATION BY DOCTOR AND NURSE USE CASE .....	45
ตารางที่ 3.9 รายละเอียดตารางผู้ใช้ (USER) .....	63
ตารางที่ 3.10 รายละเอียดตารางสิทธิ์ (ROLE) .....	64
ตารางที่ 3.11 รายละเอียดตารางสิทธิ์ (PERMISSION) .....	64
ตารางที่ 3.12 รายละเอียดตารางระบบแชต (CHAT) .....	65
ตารางที่ 3.13 รายละเอียดตารางระบบล็อก (LOG) .....	66
ตารางที่ 3.14 รายละเอียดตารางการตรวจ (PERUSAL) .....	66
ตารางที่ 3.15 รายละเอียดตารางการวินิจฉัย (DIAGNOSIS) .....	67
ตารางที่ 3.16 รายละเอียดตารางบาดแผล (WOUND) .....	67
ตารางที่ 3.17 รายละเอียดตารางระดับแผล (WOUNDSTATE) .....	68
ตารางที่ 3.18 รายละเอียดตารางวิธีการรักษา (TREAT) .....	69
ตารางที่ 4.1 ตารางเปรียบเทียบทรัพยากรของแต่ละบริการ .....	96

# สารบัญตาราง (ต่อ)

ตาราง	หน้า
ตารางที่ 4.2 ตารางแสดงรายละเอียดของชุดข้อมูลที่นำมาใช้ในการพัฒนาโมเดล สำหรับการแบ่งส่วนภาพ (SEGMENTATION MODEL) สำหรับ การทดลองครั้งที่ 1 .....	97
ตารางที่ 4.3 ตารางแสดงรายละเอียดของชุดข้อมูลที่นำมาใช้ในการพัฒนาโมเดล สำหรับการแบ่งส่วนภาพ (SEGMENTATION MODEL) สำหรับ การทดลองครั้งที่ 2 .....	98
ตารางที่ 4.4 ค่าพารามิเตอร์ที่ใช้ในการเทรน โมเดลในการทดลองรูปแบบที่ 1 .....	99
ตารางที่ 4.5 ค่าพารามิเตอร์ที่ใช้ในการเทรน โมเดลในการทดลองรูปแบบที่ 1 .....	100
ตารางที่ 4.6 ตารางวัดผลค่าแมตริกของการทดลองที่ 1 .....	101
ตารางที่ 4.7 ตารางวัดผลค่าแมตริกของการทดลองที่ 2 .....	103
ตารางที่ 4.9 ตารางวัดผลค่าแมตริกเบรย์บีนก่อนและหลังเพิ่มข้อมูลจากชุดข้อมูลแพล CLASS ที่ 1 และที่ 2 .....	108
ตารางที่ 4.10 ตารางวัดผลค่าแมตริกเบรย์บีนก่อนและหลังเพิ่มข้อมูลจากชุดข้อมูล แพลทั้งหมด .....	108
ตารางที่ 4.11 ตารางจำนวนข้อมูลแพลในแต่ละระดับ .....	110
ตารางที่ 4.12 ค่าพารามิเตอร์ที่ใช้ในการเทรน โมเดลในการทดลอง .....	111
ตารางที่ 4.13 ตารางวัดผลค่าแมตริกของการทดลองจำแนกประเภทภาระห่วง โมเดล RESNET และ EFFICIENTNET .....	112

# สารบัญ

รูป	หน้า
รูปที่ 2.1 รูปภาพภาษา JAVASCRIPT .....	3
รูปที่ 2.2 รูปภาพภาษา TYPESCRIPT .....	3
รูปที่ 2.3 รูปภาพภาษา DART .....	4
รูปที่ 2.4 รูปภาพภาษา PYTHON.....	4
รูปที่ 2.5 เฟรมเวิร์ก NESTJS.....	5
รูปที่ 2.6 เฟรมเวิร์ก ANGULAR.....	5
รูปที่ 2.7 เฟรมเวิร์ก FLUTTER.....	6
รูปที่ 2.8 เฟรมเวิร์ก PYTORCH.....	6
รูปที่ 2.9 เฟรมเวิร์ก PYTORCH.....	7
รูปที่ 2.10 เฟรมเวิร์ก PANDAS .....	7
รูปที่ 2.11 เฟรมเวิร์ก SCIKIT-LEARN .....	8
รูปที่ 2.12 เฟรมเวิร์ก REST API.....	8
รูปที่ 2.13 เฟรมเวิร์ก POSTGRES DB .....	9
รูปที่ 2.14 เฟรมเวิร์ก GOOGLE COLAB .....	9
รูปที่ 2.15 เฟรมเวิร์ก ANDROID STUDIO .....	10
รูปที่ 2.16 เฟรมเวิร์ก FASTAPI.....	10
รูปที่ 2.17 CUDA (COMPUTE UNIFIED DEVICE ARCHITECTURE).....	11
รูปที่ 2.18 OPENVPN.....	11
รูปที่ 2.19 DOCKER.....	12
รูปที่ 2.20 GITHUB .....	12
รูปที่ 2.21 DOCKER HUB .....	13
รูปที่ 2.22 NGINX (ENGINE-X) .....	13
รูปที่ 2.23 CLOUDFLARE TUNNEL.....	14
รูปที่ 2.24 KUBERNETES (K8S) .....	14
รูปที่ 2.25 รูปภาพแบบจำลองของ U-NET .....	16
รูปที่ 2.26 รูปภาพแบบจำลองของ U-NET .....	18
รูปที่ 2.28 รูปภาพแบบจำลองของ EFFICIENTNET .....	22
รูปที่ 2.29 รูปภาพกระบวนการทำงานของ DEEPSKIN MODEL.....	25

# สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.30 รูปภาพตัวอย่างของ WOUND-SEGMENTATION MODEL.....	25
รูปที่ 2.32 รูปกระบวนการทดลองในงานวิจัย DEEP LEARNING FOR AUTOMATED WOUND CLASSIFICATION AND SEGMENTATION .....	30
รูปที่ 3.1 ภาพการออกแบบของระบบซอฟต์แวร์โดยรวม .....	35
รูปที่ 3.2 ภาพการออกแบบสถาปัตยกรรมระบบซอฟต์แวร์ในรูปแบบ 3-TIER .....	36
รูปที่ 3.3 USE CASE DIAGRAM ของผู้ป่วย PATIENT .....	37
รูปที่ 3.4 USE CASE DIAGRAM ของพยาบาล NURSE .....	37
รูปที่ 3.5 USE CASE DIAGRAM ของ ERP SYSTEM.....	38
รูปที่ 3.6 ลำดับการทำงานในการลงทะเบียนผู้ใช้สำหรับผู้ป่วย .....	54
รูปที่ 3.8 ลำดับการทำงานในระบบการแพลตฟอร์มผู้ใช้งานระบบ .....	56
รูปที่ 3.9 ลำดับการทำงานในระบบการเพิ่มแพลเก็จสู่ระบบเพื่อแสดงผลการทำนาย .....	56
รูปที่ 3.10 ลำดับการทำงานในระบบติดตามอาการของผู้ป่วย .....	57
รูปที่ 3.11 ลำดับการทำงานในระบบติดตามอาการของผู้ป่วยจากพยาบาล .....	58
รูปที่ 3.12 ลำดับการทำงานในระบบการตรวจสอบความถูกต้องแพลโดยพยาบาล .....	59
รูปที่ 3.13 ลำดับการทำงานในระบบ ERP .....	60
รูปที่ 3.14 ลำดับการทำงานในระบบส่ง OTP เพื่อกรณีลืมรหัสผ่าน .....	61
รูปที่ 3.15 ภาพ ERD DIAGRAM.....	62
รูปที่ 3.16 หน้าแนะนำแอปพลิเคชัน หน้าเข้าสู่ระบบ สมัครสมาชิก และหน้าเงื่อนไข การใช้งานแอปพลิเคชัน .....	69
รูปที่ 3.17 หน้าระบบลืมรหัสและการกู้คืนรหัสผ่าน .....	70
รูปที่ 3.18 หน้าระบบการทำรายการตรวจแพล .....	71
รูปที่ 3.19 หน้ารายการแพลของในการตรวจแต่ละครั้ง.....	71
รูปที่ 3.20 หน้าระบบแพลของผู้ใช้งานทั้งผู้ป่วย หมอยา และพยาบาล .....	72
รูปที่ 3.21 หน้าระบบการติดตามแพลของผู้ใช้งานทั่วไป (ผู้ป่วย).....	73
รูปที่ 3.22 หน้าระบบการเพิ่มผู้ป่วยในการดูแลของพยาบาล .....	74
รูปที่ 3.23 หน้าระบบการบันทึกความถูกต้องของแพล .....	74
รูปที่ 3.24 หน้าระบบแพลของพยาบาล.....	75
รูปที่ 4.1 รูปผลการทดลองการใช้ระบบลงทะเบียน .....	76

# สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.2 รูปผลการทดลองการเข้าสู่ระบบ .....	77
รูปที่ 4.3 รูปผลการทดลองแสดงผลໂປຣໄຟລ໌.....	78
รูปที่ 4.4 รูปผลการทดสอบเบื้องต้นของระบบแซฟ .....	78
รูปที่ 4.5 รูปผลการทดสอบการจัดการ OTP และการเปลี่ยนรหัสผ่าน .....	79
รูปที่ 4.6 รูปผลการทดสอบการจัดการรายการตรวจสอบและรายการแพล .....	80
รูปที่ 4.7 รูปผลการทดสอบการแสดงผลการติดตามแพล .....	80
รูปที่ 4.8 รูปผลการทดสอบการแสดงผลการวินิจฉัย.....	81
รูปที่ 4.9 หน้าเข้าสู่ระบบสำหรับผู้ดูแลระบบและพยาบาล .....	82
รูปที่ 4.10 หน้าแดชบอร์ดสำหรับดูข้อมูลแพลและจำนวนผู้ป่วยในระบบ.....	82
รูปที่ 4.11 หน้าของผู้ดูแลระบบที่ใช้ในการจัดการข้อมูลคนไข้.....	83
รูปที่ 4.12 หน้าของผู้ดูแลระบบที่ใช้ในการจัดการข้อมูลพยาบาล .....	83
รูปที่ 4.13 หน้าของผู้ดูแลระบบที่ใช้ในแก้ไขข้อมูลพยาบาล.....	84
รูปที่ 4.14 หน้าของผู้ดูแลระบบที่ใช้ในแก้ไขข้อมูลผู้ป่วย .....	84
รูปที่ 4.15 หน้าของผู้ดูแลระบบที่ใช้ในการเพิ่มข้อมูลผู้ป่วย.....	85
รูปที่ 4.16 หน้าของผู้ดูแลระบบที่ใช้ในการเพิ่มข้อมูลพยาบาล .....	85
รูปที่ 4.17 หน้าของพยาบาลที่ใช้ในการดูข้อมูลผู้ป่วยในระบบ .....	86
รูปที่ 4.18 หน้า API สำหรับระบบยืนยันตัวตน .....	86
รูปที่ 4.19 หน้า API สำหรับระบบการลงทะเบียน .....	87
รูปที่ 4.20 หน้า API สำหรับตรวจสอบเลขบัตรประชาชน .....	88
รูปที่ 4.21 หน้า API สำหรับดึงข้อมูลໂປຣໄຟລ໌ของผู้ใช้.....	88
รูปที่ 4.22 หน้า API ที่เกี่ยวกับ ROLES .....	89
รูปที่ 4.23 หน้า API ที่เกี่ยวกับ PERMISSIONS .....	90
รูปที่ 4.24 หน้า API สำหรับการสร้างข้อมูลการตรวจ .....	90
รูปที่ 4.25 หน้า API สำหรับการเพิ่มข้อมูลแพล .....	91
รูปที่ 4.26 หน้า API สำหรับการแสดงข้อมูลแพล .....	91
รูปที่ 4.27 หน้า API สำหรับการจัดการระดับความรุนแรงของแพล .....	92
รูปที่ 4.28 หน้า API สำหรับการจัดการข้อมูลการรักษา .....	92
รูปที่ 4.29 หน้า API สำหรับการจัดการห้อง .....	93

## สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.30 หน้า API สำหรับการจัดการแซฟ .....	93
รูปที่ 4.31 หน้า API สำหรับการจัดการการคุ้ด .....	94
รูปที่ 4.32 หน้า API สำหรับเดชบอร์ด .....	94
รูปที่ 4.33 รูปภาพกระบวนการทำงานของระบบในส่วน โนมเดล .....	95
รูปที่ 4.34 ตัวอย่างรูปภาพแพลและหน้ากากของแพลสำหรับการทดลองครั้งที่ 1 .....	97
รูปที่ 4.35 ตัวอย่างรูปภาพแพลและหน้ากากของแพลสำหรับการทดลองครั้งที่ 2 .....	98
รูปที่ 4.36 กราฟแท่งเปรียบค่าแมตริกของการทดลองที่ 1 .....	101
รูปที่ 4.37 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ MOBILENETV2 กับ รูปทดสอบ .....	101
รูปที่ 4.38 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ MOBILENETV2 กับ รูปทดสอบ .....	102
รูปที่ 4.39 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-NET 10 EPOCHS ที่ไม่มี การเพิ่มจำนวนกับ รูปทดสอบ .....	103
รูปที่ 4.40 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-NET 30 EPOCHS ที่ไม่มี การเพิ่มจำนวนกับ รูปทดสอบ .....	103
รูปที่ 4.41 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-NET 50 EPOCHS ที่ไม่มี การเพิ่มจำนวนกับ รูปทดสอบ .....	104
รูปที่ 4.42 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-NET 10 EPOCHS ที่มี การเพิ่มจำนวนกับ รูปทดสอบ .....	104
รูปที่ 4.43 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-NET 30 EPOCHS ที่มี การเพิ่มจำนวนกับ รูปทดสอบ .....	104
รูปที่ 4.44 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-NET 50 EPOCHS ที่มี การเพิ่มจำนวนกับ รูปทดสอบ .....	105
รูปที่ 4.45 กราฟแท่งเปรียบค่าแมตริกของการทดลองที่ 2 .....	105
รูปที่ 4.46 ผลลัพธ์การทำนายของ U-NET 50 EPOCHS (AUGMENT) กับรูปแพลใน CLASS ที่ 1 .....	106
รูปที่ 4.47 ผลลัพธ์การทำนายของ U-NET 50 EPOCHS (AUGMENT) กับรูปแพลใน CLASS ที่ 2 .....	106
รูปที่ 4.48 ตัวอย่าง LABEL ที่มีการจัดทำขึ้น โดยผู้เชี่ยวชาญ .....	107
รูปที่ 4.49 ภาพของรูปแพลใน CLASS ที่ 1 และ 2 ก่อนเพิ่มข้อมูล .....	107

## สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.50 ภาพของรูปแพลใน CLASS ที่ 1 และ 2 หลังเพิ่มข้อมูล.....	108
รูปที่ 4.51 ตัวอย่างรูปแพลที่นำໄไปฝึกสอนโน้ดเดลสำหรับจำแนกประเภทภาพ.....	110
รูปที่ 4.52 กราฟแท่งเปรียบค่าแมตริกของการทดลองจำแนกประเภทภาระห่วงโน้ด RESNET และ EFFICIENTNET.....	112
รูปที่ 4.53 รูปภาพ DOCKER IMAGES ใน DOCKER HUB .....	113
รูปที่ 4.54 ภาพการทำงานของ POD และ SERVICE ใน KUBERNETES SERVER.....	114
รูปที่ 4.55 รูปภาพ PUBLIC HOSTNAME ที่มีการตั้งค่าไว้ที่ CLOUD FLARE .....	114

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาของปัญหา

ในปัจจุบันที่อัตราการเกิดของประชากรที่น้อยลงเรื่อยๆ จนเข้าใกล้สูงอายุ และปัญหาที่เกิดขึ้นส่วนใหญ่ของผู้สูงอายุคือ การเกิดบาดแผล ซึ่งการดูแลบาดแผลที่ไม่ถูกต้องเป็นสาเหตุที่ทำให้แพลงหายช้า และติดเชื้อ ซึ่งเป็นสาเหตุที่ทำให้เกิดแพลงเรื้อรังในที่สุด โดยแพลงเรื้อรังนั้นถือเป็นบาดแผลที่ไม่สามารถหายได้เองตามระยะเวลาของกระบวนการหายของแพลงปกติ เป็นแพลงที่หายช้า ไม่หาย หรือเกิดใหม่ซ้ำ แพลงเรื้อรังที่พบเห็นได้บ่อยในทุกกลุ่มอายุคือ แพลงกดทับ (Pressure injury or Pressure sore) ซึ่งแพลงดังกล่าวเป็นแพลงที่ต้องการได้รับการดูแลอย่างเหมาะสมและถูกต้อง

ดังนั้นเพื่อแก้ปัญหาดังกล่าว ทีมผู้จัดทำจึงจัดทำแอปพลิเคชัน DECUBITUS ขึ้นมาเพื่อที่จะช่วยให้ผู้ใช้งานสามารถจำแนกแพลงกดทับ เรียนรู้วิธีการดูแลรักษาแพลงกดทับเบื้องต้น และเป็นตัวช่วยในการหาแนวทางการบรรเทาแพลงสำหรับผู้ป่วย ได้โดยภายในแอปพลิเคชันจะมีการบอกถึงรายละเอียดของแพลง วิธีการดูแลแพลงเบื้องต้น และการติดตามอาการของแพลงจากทางแพทย์และพยาบาล

สุดท้ายแอปพลิเคชันนี้ยังเป็นตัวช่วยในการตัดสินใจของแพทย์และพยาบาลเพื่อใช้ในการดูแลคนไข้ที่มีแพลงกดทับ โดยภายในแอปพลิเคชันมีการเก็บข้อมูลแพลงกดทับผู้ป่วยและจำแนกประเภท ไว้เพื่อที่จะทำให้ง่ายต่อการนำเสนอปรึกษาต่อ และยังช่วยให้ตัดสินใจในการให้รักษาแพลงกดทับได้อย่างถูกต้อง รวดเร็วและแม่นยำ

#### 1.2 วัตถุประสงค์ของโครงงาน

- เพื่อพัฒนารูปแบบการดูแลผู้ป่วยที่มีแพลงกดทับ โดยใช้แอปพลิเคชัน DECUBITUS
- เพื่อพัฒนาโมเดลที่ใช้ในการคำนวณระดับความรุนแรงของแพลงกดทับ
- เพื่อพัฒนาแอปพลิเคชันที่ช่วยในการลดภาระงานในหน่วยงานบริการทางการแพทย์
- ช่วยให้ผู้ดูแลผู้ป่วยสามารถดูแลผู้ป่วยที่มีแพลงกดทับได้อย่างมีประสิทธิภาพมากขึ้น และช่วยลดเวลาในการให้บริการ

### 1.3 ขอบเขตของโครงการ

1. ระบบนี้ถูกออกแบบมาเพื่อใช้งานระหว่างคืนไชกับพยาบาลเท่านั้นทำให้ขาดการต่อ กับส่วนที่เป็นหน้าที่การทำงานของแพทย์หรือส่วนงานอื่นๆ
2. โมเดลที่ใช้ในการทำงานายแผลยังมีความถูกต้องและแม่นยำไม่มากพอ
3. ตัวแอปพลิเคชันเป็นเพียงตัวช่วยในการดูแลแผลกดทับทำให้ไม่ครอบคลุมถึงวิธีการรักษาแผลกดทับ
4. ระบบนี้ไม่ครอบคลุมถึงการวินิจฉัยทางการแพทย์
5. ระบบนี้ไม่สามารถ แทนที่การดูแลจากบุคลากรทางการแพทย์โดยตรง แต่เป็นเครื่องมือช่วยเสริมเท่านั้น
6. ระบบไม่สามารถรักษาแผลกดทับได้ แต่เป็นเพียง เครื่องมือช่วยป้องกันและแจ้งเตือน
7. การแยกระดับแผลกดทับอาจมีความคลาดเคลื่อนขึ้นอยู่กับตำแหน่งการถ่ายรูป หรือแสงของรูปภาพ

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

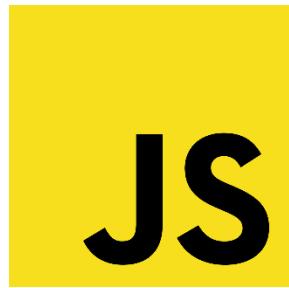
แอปพลิเคชันนี้มุ่งเน้นการดูแลแผลกดทับ ได้จากทุกที่ทุกเวลาเพื่อเป็นตัวช่วยในการลดภาระงานของบุคลากรทางการแพทย์ โดยใช้เทคโนโลยีการแพทย์ทางไกลไปกับโมเดลที่ใช้ในการช่วยทำงานายระดับของแผลและนำวิธีการดูแลแผลเบื้องต้น ออกจากนั้นยังเป็นตัวอย่างการทำโมเดลเพื่อใช้ในการจำแนกประเภทถูกอีกด้วย

## บทที่ 2

# ทฤษฎีและเครื่องมือที่เกี่ยวข้อง

### 2.1 เครื่องมือที่ใช้ในโครงการ

#### 2.1.1 JavaScript



รูปที่ 2.1 รูปภาพภาษา JavaScript

JavaScript เป็นภาษาโปรแกรมเชิงวัตถุแบบ dynamic typing ที่ได้รับการพัฒนาโดย Brendan Eich ในปี ค.ศ. 1995 ภาษาชนี้มีบทบาทสำคัญในการพัฒนาเว็บแอปพลิเคชันสมัยใหม่ โดยสามารถทำงานได้ทั้งในส่วนของ client-side และ server-side การออกแบบของ JavaScript มุ่งเน้นการสร้างปฏิสัมพันธ์แบบ asynchronous และการจัดการ Document Object Model (DOM) ซึ่งทำให้สามารถสร้างเว็บไซต์แบบ dynamic ได้อย่างมีประสิทธิภาพ นอกจากนี้ JavaScript ยังมีระบบอนิเมชันที่กว้างขวาง ประกอบด้วย ไลบรารีและเฟรมเวิร์กจำนวนมาก ซึ่งช่วยเพิ่มประสิทธิภาพและความยืดหยุ่นในการพัฒนาซอฟต์แวร์

#### 2.1.2 Typescript



รูปที่ 2.2 รูปภาพภาษา Typescript

TypeScript เป็นภาษาโปรแกรมแบบ open-source ที่พัฒนาและดูแลโดย Microsoft Corporation ซึ่งเป็น superset ของ JavaScript โดยเพิ่มเติมระบบ static typing และคุณสมบัติของภาษาโปรแกรมเชิงวัตถุขั้นสูง ภาษาชนี้ถูกออกแบบมาเพื่อพัฒนาแอปพลิเคชันขนาดใหญ่ โดยมีจุดเด่นคือการตรวจจับข้อผิดพลาดในระหว่างการพัฒนา (compile-time) และการสนับสนุน

เครื่องมือการพัฒนา (tooling support) ที่มีประสิทธิภาพ TypeScript ช่วยเพิ่มความแข็งแกร่งของ โค้ดและความสามารถในการบำรุงรักษาระบบ โดยการคอมไพล์ TypeScript จะแปลงเป็น JavaScript ที่สามารถทำงานได้บนเว็บเบราว์เซอร์ Node.js หรือ สภาพแวดล้อมการรันไทม์อื่นๆ ที่รองรับ JavaScript

### 2.1.3 Dart



รูปที่ 2.3 รูปภาษา Dart

Dart เป็นภาษาโปรแกรมเชิงวัตถุแบบ open-source ที่พัฒนาโดย Google ซึ่งออกแบบมา เพื่อการสร้างแอปพลิเคชันแบบข้ามแพลตฟอร์มที่มีประสิทธิภาพสูง ภาษาани้มีร่วมคุณสมบัติของภาษา static-typed และ dynamic-typed !เข้าด้วยกัน โดยมีไวยากรณ์ที่คล้ายคลึงกับ C-style languages Dart สนับสนุนการเขียนโปรแกรมแบบ concurrent ผ่านกลไก isolates และมี garbage collection สำหรับการจัดการหน่วยความจำอัตโนมัติ นอกจากนี้ Dart ยังเป็นภาษาหลักในการพัฒนาแอปพลิเคชันด้วย Flutter framework ซึ่งใช้สำหรับสร้างส่วนติดต่อผู้ใช้ที่มีประสิทธิภาพสูงสำหรับอุปกรณ์ มือถือ เว็บ และเดสก์ท็อป Dart สามารถคอมไпал์เป็นโค้ดเนื้อที่ฟหรือ JavaScript เพื่อรันบน แพลตฟอร์มต่างๆ ได้อย่างยืดหยุ่น

### 2.1.4 Python



รูปที่ 2.4 รูปภาษา python

Python เป็นภาษาโปรแกรมระดับสูงแบบ interpreted ที่พัฒนาโดย Guido van Rossum และเปิดตัวครั้งแรกในปี ค.ศ. 1991 ภาษานี้มีประสิทธิภาพการอ่านแบบที่เน้นความชัดเจนและความเรียบง่ายของโค้ด (code readability) โดยใช้การเว้นวรรคเป็นส่วนสำคัญของไวยากรณ์ภาษา Python

สนับสนุนหลายกระบวนการทัศน์การเขียนโปรแกรม รวมถึงการเขียนโปรแกรมเชิงวัตถุ เชิงคำสั่ง และ เชิงฟังก์ชัน อีกทั้งยังมีระบบ dynamic typing และ automatic memory management ผ่าน garbage collection

#### 2.1.5 NestJS



รูปที่ 2.5 เฟรมเวิร์ก NestJS

NestJS คือเฟรมเวิร์กสำหรับการพัฒนาเว็บแอปพลิเคชันที่มีประสิทธิภาพและสามารถขยายได้บน Node.js โดยใช้ภาษา TypeScript ซึ่งมีโครงสร้างที่ชัดเจนและรองรับการทำ Dependency Injection ทำให้เหมาะสมสำหรับการพัฒนาแอปพลิเคชันขนาดใหญ่ NestJS สร้างขึ้นบนพื้นฐานของ Express หรือ Fastify และมอบเครื่องมือที่จำเป็นสำหรับการจัดการโปรเจกต์ที่ซับซ้อนได้อย่างมีประสิทธิภาพ

#### 2.1.6 Angular



รูปที่ 2.6 เฟรมเวิร์ก Angular

Angular คือเฟรมเวิร์กการพัฒนา Web Application แบบ Opensource ที่ใช้ภาษา TypeScript ซึ่งพัฒนาโดย Google โดยมุ่งเน้นการสร้างแอปพลิเคชันแบบหน้าเดียว (Single Page Applications - SPA) ที่มีโครงสร้างที่ชัดเจนและสามารถจัดการกับโปรเจกต์ขนาดใหญ่ได้อย่างมีประสิทธิภาพ Angular ใช้ Component Architecture ที่ช่วยให้การพัฒนาและบำรุงรักษาโค้ดทำได้ง่ายขึ้น นอกจากนี้ยังมี Feature ที่สำคัญ เช่น การทำงานร่วมกันผ่าน Dependency Injection และการจัดการเส้นทาง (Routing) ที่ช่วยเพิ่มความสะดวกในการพัฒนาแอปพลิเคชัน

### 2.1.7 Flutter



รูปที่ 2.7 เฟรมเวิร์ก Flutter

Flutter คือชุดเครื่องมือ UI ที่พัฒนาโดย Google สำหรับการสร้างแอปพลิเคชันที่สามารถ และทำงานได้รวดเร็วนบน Platform ต่าง ๆ เช่น Mobile, Web, Desktop จาก Code Baseเดียว โดย Flutter ใช้ภาษา Dart ในการพัฒนา และมีคุณสมบัติที่สำคัญ เช่น Hot Reload ที่ช่วยให้สามารถปรับแต่งและดูผลการเปลี่ยนแปลงได้ทันที นอกจากนี้ยังมีการสนับสนุนจากบริการต่าง ๆ ของ Google ทำให้เป็นเครื่องมือที่เหมาะสมสำหรับนักพัฒนาที่ต้องการสร้างแอปพลิเคชันที่มีประสิทธิภาพสูง

### 2.1.8 Pytorch



รูปที่ 2.8 เฟรมเวิร์ก Pytorch

PyTorch คือเฟรมเวิร์กการเรียนรู้เชิงลึกแบบ Opensource ที่พัฒนาโดย Facebook ซึ่งรองรับการคำนวณเชิงตัวเลขด้วย Tensor ที่มีประสิทธิภาพสูงและการเร่งความเร็วด้วย GPU PyTorch มีคุณสมบัติที่สำคัญ ตัว PyTorch มี builtin library ที่สามารถนำมาใช้ในการฝึกสอนโมเดลได้หลายรูปแบบ

### 2.1.9 NumPy



รูปที่ 2.9 เฟรมเวิร์ก Pytorch

NumPy คือ ไลบรารี โอลเพ็นซอร์สสำหรับการประมวลผลข้อมูลเชิงตัวเลขในภาษา Python ซึ่งออกแบบมาเพื่อจัดการกับอาร์เรย์หลายมิติอย่างมีประสิทธิภาพ โดยมีโครงสร้างข้อมูลหลักคือ ndarray ที่ช่วยให้การดำเนินการทางคณิตศาสตร์และการวิเคราะห์ข้อมูลเป็นไปอย่างรวดเร็ว

### 2.1.10 Pandas



รูปที่ 2.10 เฟรมเวิร์ก Pandas

Pandas คือ ไลบรารีสำหรับการวิเคราะห์ข้อมูลในภาษา Python ที่ช่วยให้การจัดการและประมวลผลข้อมูลเป็นไปอย่างง่ายดาย โดยมีโครงสร้างข้อมูลหลักคือ Data Frame ซึ่งมีลักษณะคล้ายตาราง ทำให้สามารถจัดการข้อมูลที่มีหลายมิติและประเภทได้อย่างมีประสิทธิภาพ นอกจากนี้ Pandas ยังมีฟังก์ชันสำหรับการทำความสะอาดข้อมูล, การวิเคราะห์, และการสร้างกราฟ ทำให้เป็นเครื่องมือที่สำคัญในงานด้านวิทยาศาสตร์ข้อมูลและการวิเคราะห์เชิงสถิติ

### 2.1.11 Scikit-learn



รูปที่ 2.11 เฟรมเวิร์ก Scikit-learn

Scikit-learn คือ ไลบรารี โอเพ่นซอร์สสำหรับการเรียนรู้ของเครื่องในภาษา Python ที่มีความนิยมสูง ซึ่งให้เครื่องมือและ Algorithm ที่หลากหลายสำหรับการทำงานด้าน Classification, Regression และ Dimensionality reduction โดยมีฟังก์ชันที่ใช้งานง่ายและสอดคล้องกันในการประมวลผลข้อมูลและการประมวลผลโดยเดล Scikit-learn ถูกพัฒนาบนพื้นฐานของ NumPy และ SciPy ทำให้เหมาะสมสำหรับทั้งผู้เริ่มต้นและนักวิทยาศาสตร์ข้อมูลที่มีประสบการณ์

### 2.1.12 Rest API



รูปที่ 2.12 เฟรมเวิร์ก Rest API

RESTful API คืออินเทอร์เฟซที่ใช้สำหรับการสื่อสารระหว่างแอปพลิเคชัน โดยอิงตามหลักการของ Representational State Transfer (REST) ซึ่งช่วยให้การเข้าถึงและจัดการข้อมูลผ่าน HTTP ได้อย่างมีประสิทธิภาพ โดยใช้คำสั่งหลัก เช่น GET, POST, PUT และ DELETE เพื่อดำเนินการสร้าง, อ่าน, อัปเดต และลบข้อมูลในทรัพยากรต่าง ๆ RESTful API มีลักษณะเป็น Stateless ซึ่งหมายความว่าทุกคำขอจะต้องมีข้อมูลที่จำเป็นทั้งหมดสำหรับการประมวลผล ทำให้สามารถพัฒนาและปรับปรุงได้อย่างยืดหยุ่นและง่ายดาย

### 2.1.13 Postgres DB



รูปที่ 2.13 เฟรมเวิร์ก Postgres DB

PostgreSQL คือ Object-Oriented Databases แบบ Opensource ที่มีความสามารถสูง ซึ่งพัฒนามากกว่า 35 ปี โดยรองรับการจัดการข้อมูลที่ซับซ้อนและมี Feature ที่หลากหลาย เช่น การทำ ACID Transactions การทำงานร่วมกันผ่าน SQL ที่ขยายได้ ทำให้เหมาะสมสำหรับการใช้งานในแอปพลิเคชันที่ต้องการความเชื่อถือได้และประสิทธิภาพสูง

### 2.1.14 Google Colab



รูปที่ 2.14 เฟรมเวิร์ก Google Colab

Google Colab คือ Online Platform ที่ให้บริการ Jupyter Notebooks โดยไม่ต้องติดตั้งซอฟต์แวร์ใด ๆ ผู้ใช้สามารถเขียนและรันโค้ด Python ได้อย่างสะดวกสบาย พร้อมทั้งเข้าถึงทรัพยากรการประมวลผลพรี เซ็น GPU และ TPU ซึ่งเหมาะสมสำหรับการพัฒนาโมเดล Machine Learning และการวิเคราะห์ข้อมูล นอกจากนี้ยังมี Feature การทำงานร่วมกันแบบ Real-Time ทำให้ผู้ใช้สามารถแชร์โน๊ตบุ๊กและทำงานร่วมกันได้อย่างมีประสิทธิภาพในทีม

### 2.1.15 Android Studio



รูปที่ 2.15 เฟรมเวิร์ก Android Studio

Android Studio เป็นสภาพแวดล้อมการพัฒนาที่ครบวงจร (IDE) สำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android ซึ่งช่วยให้นักพัฒนาสามารถเขียนโค้ด สร้าง Layout และทดสอบแอปได้อย่างมีประสิทธิภาพ โดยมี Feature ที่ช่วยเพิ่มความสะดวกในการเขียนโค้ด

### 2.1.16 เฟรมเวิร์ก FastAPI



รูปที่ 2.16 เฟรมเวิร์ก FastAPI

FastAPI เป็นเว็บเฟรมเวิร์กแบบไลต์เวทที่สร้างด้วย Python มีประสิทธิภาพสูงและรองรับการทำงานแบบ asynchronous (async) ทำให้สามารถประมวลผลได้รวดเร็วมากขึ้น เหมาะสมสำหรับการพัฒนา API ที่ต้องการการตอบสนองที่รวดเร็ว รองรับการตรวจสอบและสร้างเอกสาร API อัตโนมัติผ่าน OpenAPI และ JSON Schema ใช้งานง่ายด้วยการกำหนดเส้นทางและการตรวจสอบข้อมูลที่เรียบง่าย เหมาะกับการพัฒนาโปรเจกต์ที่ต้องการความเร็วและความยืดหยุ่น

### 2.1.17 CUDA (Compute Unified Device Architecture)



รูปที่ 2.17 CUDA (Compute Unified Device Architecture)

CUDA (Compute Unified Device Architecture) เป็นแพลตฟอร์มการประมวลผลแบบขนาน ที่พัฒนาโดย NVIDIA เพื่อให้การ์ดจอ (GPU) สามารถใช้ประมวลผลข้อมูลนอกเหนือจากกราฟิกได้ โดย CUDA ช่วยให้การ์ดจอสามารถทำงานร่วมกับ CPU ใน การประมวลผลงานหนักๆ ที่ใช้เวลาและทรัพยากรามาก เช่น การประมวลผลภาพ การคำนวณทางวิทยาศาสตร์ และการฝึกไมโครเดลปัญญาประดิษฐ์ ช่วยเพิ่มประสิทธิภาพการทำงานและลดเวลาการประมวลผล

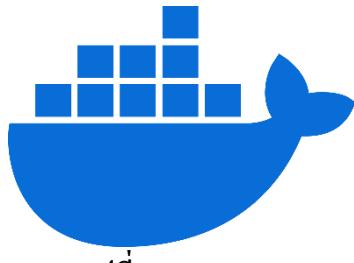
### 2.1.18 OpenVPN



รูปที่ 2.18 OpenVPN

OpenVPN คือซอฟต์แวร์ Open Source สำหรับสร้างเครือข่ายเสมือน (VPN) ที่ปลอดภัยใช้เพื่อเข้ารหัสการเชื่อมต่อระหว่างอุปกรณ์และเซิร์ฟเวอร์ ช่วยปกป้องข้อมูลและความเป็นส่วนตัวรองรับการตั้งค่าทั้งแบบ Site-to-Site และ Remote Access สามารถใช้งานได้หลายแพลตฟอร์ม เช่น Windows macOS Linux รวมถึงอุปกรณ์มือถือ

### 2.1.19 Docker



รูปที่ 2.19 Docker

Docker คือแพลตฟอร์มที่ช่วยในการพัฒนา ส่งมอบ และรันแอปพลิเคชันในรูปแบบ Container โดย Container เป็นหน่วยเล็ก ๆ ที่รวมทุกสิ่งที่แอปพลิเคชันต้องการ เช่น โค้ด ไลบรารี และการตั้งค่าต่างๆ ทำให้สามารถรันแอปพลิเคชันได้บนทุกสภาพแวดล้อม ไม่ว่าจะเป็นเครื่อง Local, Server หรือ Cloud

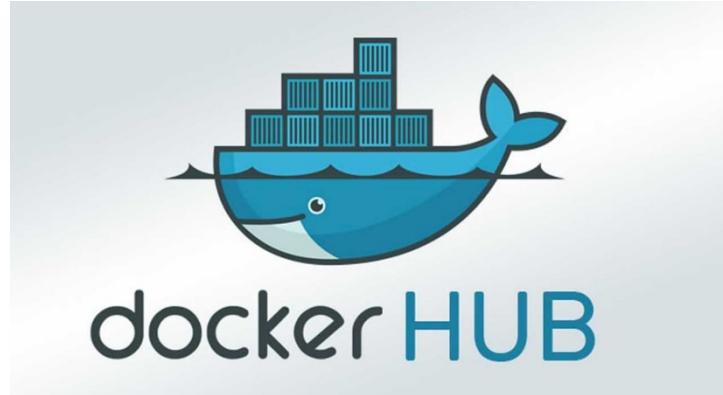
### 2.1.20 Github



รูปที่ 2.20 Github

GitHub คือแพลตฟอร์มสำหรับจัดการโค้ดโปรเจกต์ที่ใช้ Git ในการควบคุมเวอร์ชัน (Version Control System) ช่วยให้ทีมพัฒนาสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ โดย GitHub ทำหน้าที่เป็นพื้นที่เก็บโค้ดแบบออนไลน์ (Repository) ที่รองรับทั้งโปรเจกต์แบบส่วนตัว และแบบสาธารณะ

### 2.1.21 Docker Hub



รูปที่ 2.21 Docker Hub

Docker Hub คือบริการออนไลน์สำหรับเก็บและแชร์ Docker Images ทำหน้าที่เหมือนคลังเก็บภาพที่ช่วยให้นักพัฒนาและทีมสามารถอัปโหลด ดาวน์โหลด และแชร์ Images ได้อย่างสะดวก มีฟีเจอร์สำคัญ เช่น Image Repository สำหรับเก็บ Images ทั้งแบบสาธารณะและส่วนตัว Prebuilt Images

### 2.1.22 Nginx (Engine-X)



รูปที่ 2.22 Nginx (Engine-X)

Nginx (Engine-X) คือเว็บเซิร์ฟเวอร์ความเร็วสูงที่ใช้สำหรับให้บริการเว็บไซต์, ทำ Reverse Proxy, โหลดбалานซ์ (Load Balancer) และพร็อกซีเซิร์ฟเวอร์ โดยออกแบบมาให้รองรับการเชื่อมต่อจำนวนมากด้วยการประมวลผลแบบ Asynchronous มีจุดเด่นคือความเร็ว ใช้ทรัพยากรน้อย และรองรับทั้ง Static และ Dynamic Content เหมาะสำหรับการโอนสต็อคเร็วๆ

### 2.1.23 Cloudflare Tunnel



รูปที่ 2.23 Cloudflare Tunnel

Cloudflare Tunnel คือบริการที่ช่วยสร้างการเชื่อมต่อแบบปลอดภัยระหว่างเซิร์ฟเวอร์หรือแอปพลิเคชันในเครื่องของคุณกับอินเทอร์เน็ต โดยไม่ต้องเปิดพอร์ตหรือกำหนดค่าไฟร์วอลล์ทำงานผ่านproto協議ของ Cloudflare เพื่อให้บริการอินเทอร์เน็ตเข้าถึงเซิร์ฟเวอร์ที่อยู่เบื้องหลังได้อย่างปลอดภัย

### 2.1.24 Kubernetes (K8s)



รูปที่ 2.24 Kubernetes (K8s)

Kubernetes (K8s) เป็นแพลตฟอร์มจัดการคอนเทนเนอร์ที่ช่วยในการ ปรับใช้ (Deploy), ปรับขนาด (Scale) และ จัดการ (Manage) แอปพลิเคชันแบบอัตโนมัติ รองรับการทำงานแบบกระจายโหลด (Load Balancing), การรีสีนตัวเอง (Self-healing), และการจัดการการอัปเดต (Rolling Updates) ทำให้การบริหารแอปพลิเคชันที่ใช้คอนเทนเนอร์มีประสิทธิภาพและยืดหยุ่น

## 2.2 ทฤษฎีที่ใช้งานวิจัย

### 2.2.1 Edge Detection

Edge Detection เป็นเทคนิคพื้นฐานในการประมวลผลภาพดิจิทัลที่ใช้สำหรับการค้นหา และระบุขอบเขตของวัตถุภายในภาพการตรวจจับขอบเป็นขั้นตอนสำคัญในหลายกระบวนการที่เกี่ยวข้องกับการวิเคราะห์ภาพและการมองเห็นของคอมพิวเตอร์

หลักการของ Edge Detection ทำงานโดยการค้นหาจุดในภาพที่มีการเปลี่ยนแปลงความเข้มแสงอย่างชัดเจนและการเปลี่ยนแปลงนี้สามารถตรวจจับได้โดยการคำนวณ Gradient ของความเข้มแสงภาพในแต่ละจุดของภาพ เพื่อหาขอบเขตระหว่างวัตถุและพื้นหลังของภาพ

### 2.2.2 Texture Analysis

Texture Analysis เป็นการวิเคราะห์ลักษณะที่เกี่ยวข้องกับเนื้อสัมผัสของวัตถุหรือพื้นผิวในภาพดิจิทัลซึ่งเป็นหนึ่งในคุณลักษณะที่สำคัญและมีประโยชน์ในการวิเคราะห์ภาพและการมองเห็นของคอมพิวเตอร์ โดยจะวิเคราะห์ถึงรูปแบบ ความหนาแน่น ความหมาย และความละเอียดที่มีอยู่ในพื้นผิว

หลักการของ Texture Analysis จะมีการระบุและการแยกแยะเนื้อสัมผัสโดยการคำนวณคุณสมบัติต่างๆ จากพื้นผิว ซึ่งรวมถึงรูปแบบของการกระจายแสง ความเข้ม และความสัมพันธ์ระหว่างพิกเซลต่างๆ ในภาพแล้วนำค่าที่คำนวณได้มาจำแนกประเภทเนื้อสัมผัสที่มีคุณสมบัติที่ได้จากการวิเคราะห์เนื้อสัมผัสเพื่อจำแนกประเภทของเนื้อสัมผัส โดยเปรียบเทียบกับเนื้อสัมผัสอื่นๆ ที่มีลักษณะคล้ายคลึงกัน

### 2.2.3 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) เป็นหนึ่งในโมเดลประดิษฐ์ Deep Learning ที่มีประสิทธิภาพสูงสำหรับงานที่เกี่ยวข้องกับการมองเห็นของคอมพิวเตอร์ เช่น การจำแนกภาพ การตรวจจับวัตถุ การจำแนกใบหน้า และอื่นๆ โครงสร้างของ CNNs ถูกออกแบบมาเพื่อจำลองวิธีที่มนุษย์มองเห็นและรับรู้ภาพ โดยมีลักษณะการทำงานแบ่งเป็นชั้น ได้ดังนี้

- Convolutional Layers: เป็นหัวใจของ CNNs ทำหน้าที่กรอง (filter) ข้อมูลจากภาพเพื่อกรองลักษณะเฉพาะ (features) ออกมานอก ขอบ, รูปทรง, และพื้นผิว
- Pooling Layers: ใช้สำหรับลดขนาดของข้อมูลที่ผ่านมาจาก convolutional layers เพื่อลดปริมาณการคำนวณและช่วยในการแยกแยะลักษณะเฉพาะที่สำคัญๆ
- FullyConnectedLayers: ในชั้นนี้มีหน้าที่สำคัญในการทำนายผลจากคุณลักษณะ(features)ที่ได้รับการสรุปจากชั้น convolutional และ pooling layers

#### 2.2.4 Wound Segmentation

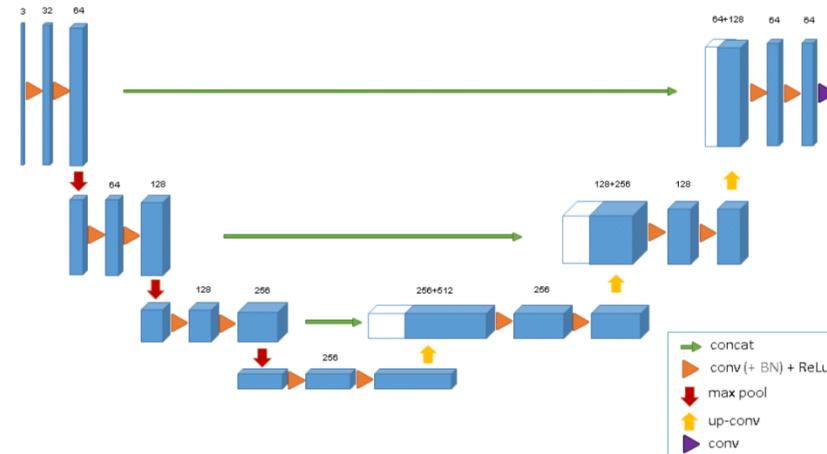
Wound Segmentation เป็นกระบวนการซึ่งมีเป้าหมายเพื่อระบุและแยกบาดแผลออกจากส่วนอื่นๆ ของภาพ. กระบวนการนี้มีหลายขั้นตอนและใช้เทคนิคต่างๆ เพื่อประมวลผลและวิเคราะห์ภาพ กระบวนการนี้มีความสำคัญอย่างมากเพื่อให้ตัวโน้มเดลสามารถนำรีเวนภาพของแผลที่ได้ไปใช้วิเคราะห์

#### 2.2.5 Wound Classification

Wound Classification เป็นกระบวนการที่สำคัญซึ่งช่วยให้สามารถประเมินและกำหนดแนวทางการรักษาที่เหมาะสมได้. วิธีการที่ใช้ในการแยกและนิยดบาดแผลสามารถแบ่งออกเป็นหลายรูปแบบ ตั้งแต่การประเมินโดยตรงจากแพทย์หรือบุคลากรทางการแพทย์ไปจนถึงการใช้เทคโนโลยีสมัยใหม่ เช่น Machine Learning และ Deep Learning

#### 2.2.6 U-net

U-net เป็นโน้มเดล Convolutional Neural Network (CNNs) ที่ถูกออกแบบมาเพื่องาน Image Segmentation โดยเฉพาะ ซึ่งทำงานโดยการแบ่งส่วนภาพเพื่อระบุตำแหน่งของวัตถุในภาพ จุดเด่นของ U-net คือสามารถทำงานได้ดีกับข้อมูลภาพทางการแพทย์ซึ่งมีข้อมูลที่จำกัดและต้องการความแม่นยำสูงในการแบ่งส่วนวัตถุ ด้วยกระบวนการทำงานที่มีการเชื่อมต่อขั้นชั้น (skip connections) ช่วยให้การเรียนรู้และการทำงานมีความแม่นยำที่สูง โดยโครงสร้างของ U-net มีลักษณะเป็นรูปตัว U สามารถแบ่งออกมาเป็นชั้น ได้ดังนี้



รูปที่ 2.25 รูปภาพแบบจำลองของ U-net

##### 2.2.6.1 ส่วนย่อ (Contracting path) หรือ Encoder

ส่วนนี้เป็นทางด้านซ้ายของ U ซึ่งเป็นเส้นทางการหดตัวมีการใช้ Convolutional layers และ Max pooling layers เพื่อจับคุณลักษณะ (features) ของภาพในขณะที่ลดขนาดมิติของภาพลง โดยสามารถแบ่งรายละเอียดการทำงานของแต่ละชั้นได้ดังนี้

### 2.2.6.1.1 Convolutional Layers

ในส่วน Encoder ขั้นนี้ทำการสกัดคุณลักษณะจากข้อมูลภาพเข้าโดยใช้เทคนิคที่เรียกว่า "convolution" ซึ่งเป็นการคูณและรวมค่า (summing) ระหว่าง weights ของ filter กับค่า pixel ซึ่งการทำงานของขั้น Convolutional ในส่วนของ Encoder ประกอบด้วยการใช้งานสามองค์ประกอบหลัก ได้แก่ Filters/Kernels, Strides, และ Padding ซึ่งมีรายละเอียดดังนี้

- Filters/Kernels เมทริกขนาดเล็ก (เช่น 3x3 หรือ 5x5) ที่มีค่า weights ซึ่งสามารถปรับเปลี่ยนได้ตามกระบวนการเรียนรู้ของโมเดล ในระหว่างการฝึกอบรม โมเดลจะเรียนรู้ค่า weights ที่เหมาะสมซึ่งช่วยให้สามารถสกัดคุณลักษณะสำคัญจากภาพได้
- Strides คือ พารามิเตอร์ที่กำหนดขนาดของการเลื่อนของ filter บนภาพ ซึ่งระบุว่า filter จะเลื่อนไปยังตำแหน่งถัดไปบนภาพด้วยระยะทางเท่าใด การตั้งค่าของ strides มีผลต่อขนาดของ feature map ที่สร้างขึ้น

### 2.2.6.1.2 Padding

คือ ทำการลดขนาดของ feature maps โดยการสรุปค่าในพื้นที่ที่กำหนดใน feature map ด้วยการเลือกค่าสูงสุด (max pooling) หรือค่าเฉลี่ย (average pooling) จากพื้นที่นั้นๆ กระบวนการนี้จะทำให้ได้ feature map ขนาดเล็กลงที่บังคับเน้นคุณลักษณะสำคัญ

### 2.2.6.2 ส่วนขยาย (Expansive path) หรือ Decoder:

ส่วนนี้เป็นทางด้านขวาของ U ซึ่งเป็นเส้นทางการขยายที่ใช้ Convolutional layers และ Up-sampling layers เพื่อเพิ่มขนาดมิติของภาพกลับคืนมาพร้อมกับการรวมข้อมูลคุณลักษณะที่เรียนรู้จากส่วนย่อเข้ากับเส้นทางส่วนขยายการรวมคุณลักษณะนี้ช่วยให้โมเดลสามารถทำงานอย่างมีประสิทธิภาพ แต่ต้องมีข้อจำกัดคือต้องมีข้อมูลทางเดียวที่สามารถบันทึกและจัดการได้

#### 2.2.6.2.1 Upsampling Layers

ขั้นนี้มีหน้าที่ในการเพิ่มขนาด spatial ของ feature maps เทคนิคที่ใช้รวมถึงการใช้ transposed convolution (หรือ convolutional up-sampling) และ nearest neighbor up sampling กระบวนการนี้ช่วยในการฟื้นฟูขนาด spatial ของข้อมูลให้ใกล้เคียงหรือเท่ากับขนาดเดิมของภาพ

#### 2.2.6.2.2 Convolutional Layers

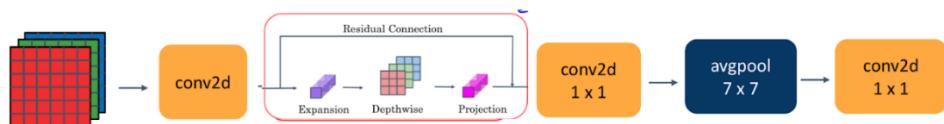
ในส่วน Decoder ขั้นนี้มีหน้าที่สำคัญในการฟื้นฟูและปรับปรุงคุณลักษณะของข้อมูลหลังจากการ upsampling และการรวมข้อมูลจาก encoder ผ่านการใช้ skip connection การปรับปรุงคุณภาพของ feature maps และการรักษารายละเอียดเชิงพื้นที่จากการเดิม เป็นสิ่งสำคัญในการทำให้โมเดลสามารถทำการ segmentation ที่มีความละเอียดและแม่นยำสูง

### 2.2.6.2.3 Skip Connections

ในขั้นนี้จะใช้เทคนิคการรวมข้อมูลจาก layers ในส่วนของ Encoder ที่มีขนาดเท่ากันมาเข้า Decoder ผ่านการ concatenate กับ feature maps ที่ได้จากการ upsampling กระบวนการนี้ช่วยให้โมเดลสามารถรักษาคุณลักษณะเชิงพื้นที่ที่สำคัญจากการเดิมได้

### 2.2.7 MobileNetsV2

MobileNetV2 เป็นโมเดล Convolutional Neural Network (CNNs) ที่เน้นการใช้งานทรัพยากรที่จำกัด ได้แก่ ประสิทธิภาพสูง โดยเฉพาะบนอุปกรณ์พกพา เริ่มต้นโดยมีการทำงานเริ่มต้นที่ Initial Convolutional Layer เพื่อเพิ่มความลึกของภาพเข้าด้วยค่า stride ขนาดใหญ่จากนั้นตามด้วย Bottleneck Layers ซึ่งเป็นส่วนกลางของสถาปัตยกรรมที่ใช้เทคนิคต่างๆ เพื่อสกัดคุณลักษณะที่สำคัญในขณะที่ประยัดการคำนวณไปด้วยนอกจากนี้ยังมีการใช้ Final Layers และการ Average Pooling ข้อมูลเพื่อลดมิติเชิงพื้นที่ลง ทั้งหมดนี้ทำให้ MobileNetV2 เหมาะอย่างยิ่งสำหรับการใช้งานที่ต้องการความเร็วและความแม่นยำในการประมวลผลภาพ โดยไม่ใช้ทรัพยากรากอนกินไปโดยโกรงสร้างของ MobileNetsV2 สามารถแบ่งออกมาเป็นชั้นได้ดังนี้



รูปที่ 2.26 รูปภาพแบบจำลองของ U-net

### 2.2.7.1 Initial Convolutional Layer

ขั้นนี้มีหน้าที่ในการสร้างเซตแรกของ feature maps จากภาพที่นำเข้าโดยใช้ convolution กับ kernel ขนาด  $3 \times 3$  ที่มีค่า stride ที่ 2 ซึ่งหมายถึงการย้าย kernel ไปทีละสองพิกเซล บนภาพเข้า ด้วยการทำนี้ ขั้นนี้ลดมิติเชิงพื้นที่ของภาพลงครึ่งหนึ่งในขณะที่เพิ่มจำนวนช่องสัญญาณซึ่งมักเรียกว่า depth ของ feature map

### 2.2.7.2 Bottleneck Layers

ซึ่งการทำงานที่ซับซ้อนและเป็นส่วนสำคัญของสถาปัตยกรรม ซึ่งออกแบบมาเพื่อประยัดทรัพยากรในขณะที่ยังคงคุณลักษณะสำคัญสำหรับการประมวลผล

#### 2.2.7.2.1 Expansion

ขั้นนี้ใช้ convolution ขนาด  $1 \times 1$  (Pointwise Convolution) เพื่อย้ายจำนวน channels หรือความลึกของ feature maps ทำให้มีมิติที่กว้างขึ้น ช่วยเพิ่มความสามารถในการสกัดคุณลักษณะที่ซับซ้อนก่อนที่จะถูกส่งต่อไปยังขั้นตอนถัดไป

#### **2.2.7.2.2 Depthwise Convolution**

ในชั้นนี้ convolution ถูกประยุกต์ใช้เฉพาะบนแต่ละ channel ของ feature map โดยไม่ข้ามระหว่าง channels นี้ช่วยให้สามารถสกัดคุณลักษณะเฉพาะพื้นที่แต่ละช่องสัญญาณได้โดยใช้การคำนวณที่ลดลงอย่างมาก

#### **2.2.7.2.3 Projection**

ที่ชั้นนี้ convolution ขนาด  $1 \times 1$  จะถูกใช้เพื่อลดจำนวน channels ของ feature map กลับลงเหลือขนาดเดิมหรือขนาดที่ใกล้เคียงกับขนาดเดิม สิ่งนี้ช่วยให้เครือข่ายรักษาคุณลักษณะที่สำคัญในขณะที่ยังคงลดทรัพยากรการคำนวณไว้

#### **2.2.7.3 Final Layers**

มีบทบาทสำคัญในการนำคุณลักษณะที่ได้จากชั้นก่อนหน้ามาใช้เพื่อการจำแนกประเภทหรือการตรวจจับวัตถุ

##### **2.2.7.3.1 Convolution2D**

เป็นชั้นที่มีการ convolution ขนาด  $1 \times 1$  เพื่อปรับปรุงครั้งสุดท้ายก่อนการจำแนกประเภท

##### **2.2.7.3.2 Average Pooling ขนาด $7 \times 7$**

ใช้เพื่อลดมิติของ feature maps โดยการคำนวณค่าเฉลี่ยของค่าพิกเซลในพื้นที่ที่กำหนด และสร้าง output ที่มีขนาดเล็กลงจากขนาดเดิม มีวัตถุประสงค์หลักคือการลดข้อมูลที่ไม่จำเป็น และเน้นคุณลักษณะที่สำคัญเพื่อปรับปรุงประสิทธิภาพการเรียนรู้และลดโอกาสของการเกิด overfitting ในโมเดล

##### **2.2.7.3.3 Convolution 2D $1 \times 1$ ชั้นสุดท้าย**

จะลดความลึกของ feature map ลงเพื่อทำนายคลาสของภาพจากจำนวนช่องสัญญาณเท่ากับจำนวนคลาสในชุดข้อมูล

#### **2.2.8 Supervised Learning**

Supervised Learning หรือการเรียนรู้แบบมีผู้ดูแล เป็นหนึ่งในประเภทหลักของการเรียนรู้ของเครื่อง (Machine Learning) ที่ไม่เดลคอมพิวเตอร์ถูกฝึกอบรมด้วยข้อมูลที่มีการกำกับไว้ (Labeled Data) โดยที่ข้อมูลนี้ประกอบด้วยข้อมูลที่จะนำเข้าไมโครและผลลัพธ์ที่ต้องการ เพื่อให้ไมโครเรียนรู้ที่จะทำนายผลลัพธ์จากข้อมูลใหม่ที่ไม่เคยเห็นมาก่อนได้อย่างแม่นยำ โดยมีเป้าหมายเพื่อลดข้อผิดพลาดในการทำนายและปรับปรุงความแม่นยำของโมเดล

#### **2.2.9 Intersection over Union (IOU)**

IOU เป็นเมตริกที่ใช้ในการประเมินความแม่นยำของโมเดลในการแบ่งส่วนภาพ (Image Segmentation) หรือการตรวจจับวัตถุ (Object Detection) โดยเป็นการวัดสัดส่วนของพื้นที่ที่ไม่คาดคาดการณ์ (Predicted Region) กับพื้นที่จริง (Ground Truth Region) ที่มีความทับซ้อนกัน

สูตรการคำนวณ:

$$IOU = \frac{|A \cap B|}{|A \cup B|} \quad (2.1)$$

อธิบายส่วนประกอบ:

- A คือพื้นที่ที่โมเดลคาดการณ์
- B คือพื้นที่ที่เป็นความจริง (Ground Truth)
- $|A \cap B|$  พื้นที่ที่หับซ้อนกันระหว่างสองพื้นที่
- $|A \cup B|$  พื้นที่รวมทั้งหมดของสองพื้นที่

#### 2.2.10 Dice Coefficient (Dice Similarity Coefficient)

Dice Coefficient เป็นตัวชี้วัดที่ใช้ในการวัดความคล้ายคลึงกันระหว่างสองชุดข้อมูล โดยเฉพาะในงานที่เกี่ยวข้องกับการแบ่งส่วนภาพ เช่น การแบ่งส่วนของเนื้อเยื่อในภาพทางการแพทย์

สูตรการคำนวณ:

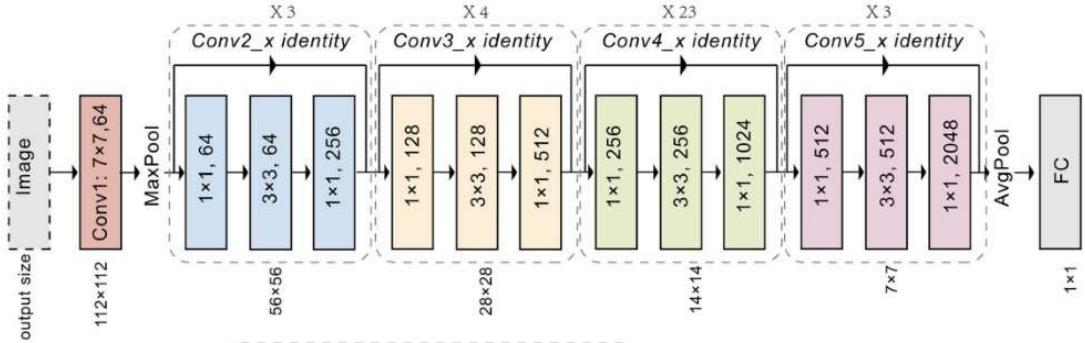
$$Dice = \frac{2 \times |A \cap B|}{|A| + |B|} \quad (2.2)$$

อธิบายส่วนประกอบ:

- A คือพื้นที่ที่โมเดลคาดการณ์
- B คือพื้นที่ที่เป็นความจริง (Ground Truth)
- $|A \cap B|$ : พื้นที่ที่หับซ้อนกันระหว่างสองพื้นที่
- $|A|$ : พื้นที่ของชุดที่โมเดลคาดการณ์
- $|B|$ : พื้นที่ของชุดความจริง

#### 2.2.11 ResNet

ResNet (Residual Network) เป็นสถาปัตยกรรมเครือข่ายแบบ Convolutional Neural Network (CNN) ที่ออกแบบมาเพื่อแก้ปัญหา vanishing gradient ใน deep networks โดยเพิ่ม skip connections เพื่อให้ข้อมูลถูกส่งข้ามเลเยอร์ได้โดยตรง โครงสร้างหลักของ ResNet มีส่วนประกอบดังนี้



รูปที่ 2.27 รูปภาพแบบจำลองของ ResNet

### 2.2.11.1 Input Layer

เมื่อรูปภาพถูกป้อนเข้าสู่เครือข่าย ResNet ภาพนี้ถูกตัดความเป็นข้อมูลคิจทั้งหมดออกไปด้วย 3 ช่องทาง (RGB) ซึ่งจะถูกแปลงและผ่านกระบวนการต่างๆ เพื่อเดิงไฟเบอร์อุกมา

### 2.2.11.2 Conv1 Layer ขั้นแรก

ภาพนี้จะผ่านชั้น convolution ขนาด  $7 \times 7$  พร้อม stride 2 ซึ่งจะช่วยลดขนาดของภาพเป็น  $112 \times 112$  แต่ขยายจำนวน channel เป็น 64 เพื่อเพิ่มมิติของข้อมูลที่ได้จากการแยกน้ำตามด้วยการทำ Batch Normalization เพื่อลดการเปลี่ยนเบนของข้อมูล และ ReLU เพื่อเพิ่มความไม่เชิงเส้นในการคำนวณ สุดท้ายทำ max pooling เพื่อลดขนาดอีกครั้ง

### 2.2.11.3 Residual Blocks

ที่นี่คือหัวใจของ ResNet เมื่อภาพถูกย่อและเดิงไฟเบอร์พื้นฐานอุกมาแล้ว ข้อมูลจะถูกส่งเข้าสู่ Residual Blocks ซึ่งเป็นชุดที่สร้างความแตกต่างของ ResNet จากเครือข่ายทั่วไป โดย Residual Blocks แต่ละบล็อกจะประกอบด้วยชั้น convolution ที่เดิงไฟเบอร์เชิงลึกมากขึ้น เช่น ขอบภาพ, รูปทรง หรือ texture ของวัตถุในภาพ โดยที่สิ่งสำคัญคือการเพิ่ม skip connection ซึ่งเป็นทางลัดให้ข้อมูลจาก input ถูกส่งไปยังชั้นที่ลึกกว่าโดยตรง ทำให้ข้อมูลต้นฉบับสามารถสมมูลกับไฟเบอร์ที่ได้จาก convolution แต่ละชั้นได้ง่ายขึ้น การเชื่อมต่อแบบนี้จะช่วยลดปัญหา gradient ที่ลดลงในระหว่างการฝึกโมเดล ทำให้ไม่เกิดเรียนรู้ได้แม้จะมีเดียร์ที่ลึกมาก

### 2.2.11.4 Average Pooling Layer

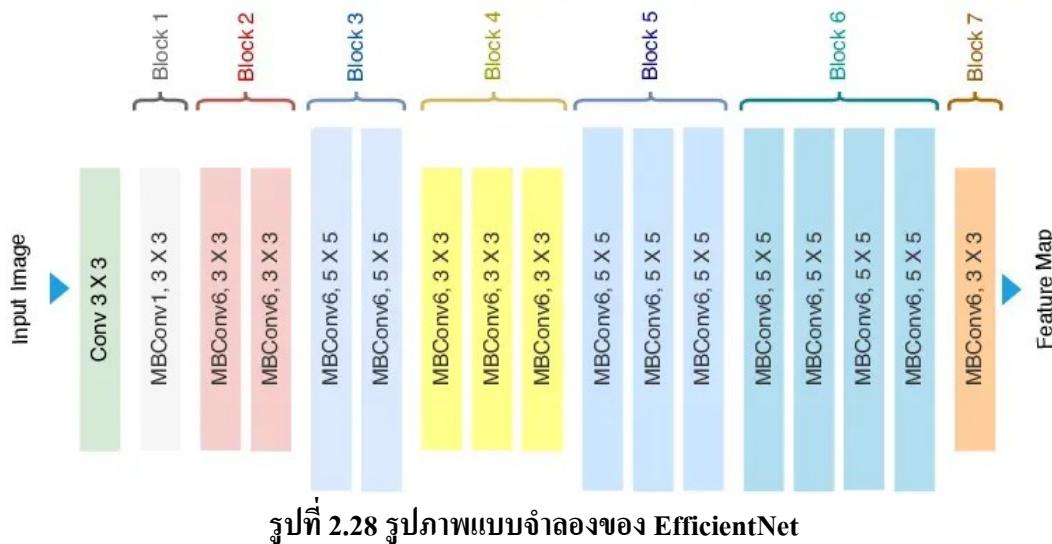
เมื่อภาพผ่านหลายเลเยอร์ของการเดิงไฟเบอร์แล้ว ระบบจะใช้ชั้น Average Pooling โดยการลดขนาดไฟเบอร์ทั้งหมดลงเป็นค่าเฉลี่ยของทุกตำแหน่ง ซึ่งช่วยลดข้อมูลเหลือเพียงขนาด  $1 \times 1 \times 512$  การทำเช่นนี้จะรักษาไฟเบอร์สำคัญสุดท้ายไว้ก่อนนำไปใช้ในการจำแนก

### 2.2.11.5 Fully Connected Layer

ไฟเบอร์ที่ได้จาก pooling จะถูกส่งต่อไปยัง Fully Connected Layer ซึ่งเป็นชั้นที่ทำหน้าที่จำแนกประเภทของภาพ เป็นคลาสที่ต้องการ

### 2.2.12 EfficientNet

EfficientNet เป็นสถาปัตยกรรมเครือข่ายประสาทเทียมที่ถูกออกแบบมาเพื่อลดความซับซ้อนในการคำนวณ แต่ยังคงความแม่นยำสูงในการจำแนกภาพ โดย EfficientNet ใช้แนวคิดในการขยายขนาดเครือข่ายอย่างมีประสิทธิภาพผ่านการเพิ่มความกว้าง (width), ความลึก (depth), และความละเอียด (resolution) อย่างสมดุล เพื่อเพิ่มประสิทธิภาพการเรียนรู้และลดการใช้ทรัพยากร



รูปที่ 2.28 รูปภาพแบบจำลองของ EfficientNet

#### 2.2.12.1 Input Layer

กระบวนการเริ่มต้นด้วยภาพ ซึ่งถูกส่งเข้ามาเป็นข้อมูล input ที่ประกอบไปด้วย 3 channels (RGB) จากนั้น EfficientNet จะทำการขยายหรือย่อขนาดภาพเพื่อให้เหมาะสมกับความละเอียดที่เครือข่ายต้องการ

#### 2.2.12.2 Stem Layer

หลังจากภาพเข้าสู่เครือข่าย จะผ่านชั้น Stem ซึ่งประกอบด้วย convolution ขนาด 3x3 ที่มี stride 2 ทำหน้าที่เหมือนชั้นแรกของ convolution ในโมเดลทั่วไป โดยทำการลดขนาดภาพลงในขณะที่เพิ่มจำนวน channel ของฟีเจอร์ ข้อมูลที่ได้จะถูกส่งไปยัง Mobile Inverted Bottleneck Conv (MBConv)

#### 2.2.12.3 Mobile Inverted Bottleneck Conv (MBConv)

ชั้น MBConv เป็นเลเยอร์สำคัญของ EfficientNet ซึ่งต่อจากแนวคิดของ MobileNet โดยเน้นการเพิ่มความแม่นยำในการดึงฟีเจอร์เชิงลึกแต่ใช้การคำนวณที่น้อยลง ซึ่งแต่ละ MBConv จะทำงานใน 3 ขั้นตอนหลัก

##### 2.2.12.3.1 Expand

ขยายจำนวน channel ขึ้นเพื่อดึงฟีเจอร์ที่ซับซ้อนยิ่งขึ้น

### 2.2.12.3.2 Depthwise Convolution

ทำ convolution กับแต่ละ channel อย่างแยกกัน ซึ่งช่วยลดการคำนวณที่ไม่จำเป็น

### 2.2.12.3.3 Project

ย่อจำนวน channel กลับไปเพื่อลดการใช้หน่วยความจำ

### 2.2.12.4 Average Pooling Layer

หลังจากภาพผ่านเลเยอร์เชิงลึก ฟีเจอร์ที่ได้จะถูกสรุปผ่าน Average Pooling เพื่อลดขนาดฟีเจอร์เหลือเพียงข้อมูลที่สำคัญที่สุดในรูปแบบ  $1 \times 1 \times$  จำนวน channel ทำให้ไม่เดลสามารถนำฟีเจอร์ไปใช้ในการจำแนกประเภทได้อย่างมีประสิทธิภาพ

### 2.2.12.5 Fully Connected Layer

ฟีเจอร์ที่ได้จาก pooling จะถูกส่งต่อไปยัง Fully Connected Layer ซึ่งเป็นชั้นที่ทำหน้าที่จำแนกประเภทของภาพ เป็นคลาสที่ต้องการ

### 2.2.13 F1 Score

F1 Score เป็นตัวชี้วัดทางสถิติที่ใช้ในการประเมินประสิทธิภาพของโมเดลการจำแนกประเภท โดยเฉพาะในกรณีที่ข้อมูลมีความไม่สมดุลระหว่างกลุ่ม (imbalanced data) ค่าที่เป็นค่าเฉลี่ยเชิง-armonik (harmonic mean) ระหว่าง Precision และ Recall ซึ่งช่วยสะท้อนถึงความสมดุลระหว่างการคาดการณ์ผลบวกที่ถูกต้องและความสามารถในการระบุผลบวกที่แท้จริงในข้อมูล

**Precision** คือ สัดส่วนของข้อมูลที่คาดการณ์ว่าเป็นบวกแล้วถูกต้องจริง ซึ่งคำนวณได้จากสูตรการคำนวณ:

$$Precision = \frac{TP}{TP + FP} \quad (2.3)$$

อธิบายส่วนประกอบ:

- TP (True Positive) คือ จำนวนผลบวกที่คาดการณ์ได้ถูกต้อง
- FP (False Positive) คือ จำนวนผลบวกที่คาดการณ์ผิด

**Recall (หรือ Sensitivity)** คือ สัดส่วนของข้อมูลบวกทั้งหมดที่ถูกคาดการณ์ออกมาระบุเป็นบวก ซึ่งคำนวณได้จาก

สูตรการคำนวณ:

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

อธิบายส่วนประกอบ:

- TP (True Positive) คือ จำนวนผลบวกที่คาดการณ์ได้ถูกต้อง
- FN (False Negative) คือ จำนวนผลบวกที่คาดการณ์ว่าเป็นลบผิดพลาด

จากนั้น F1 Score คำนวณจากค่าเฉลี่ยเชิง harmonic ของ Precision และ Recall ดังนี้

สูตรการคำนวณ:

$$F1 Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.5)$$

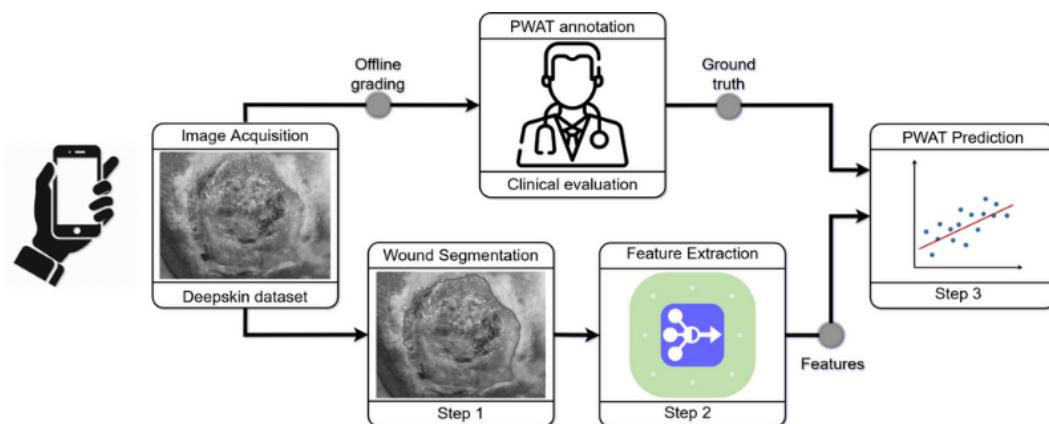
อธิบายส่วนประกอบ:

- ค่า F1 Score อยู่ในช่วง 0 ถึง 1 ซึ่งค่าใกล้ 1 หมายถึง โมเดลมีประสิทธิภาพในการคาดการณ์ที่สูง

## 2.3 งานวิจัยที่เกี่ยวข้อง

### 2.3.1 Deepskin Automated Wound Image Segmentation

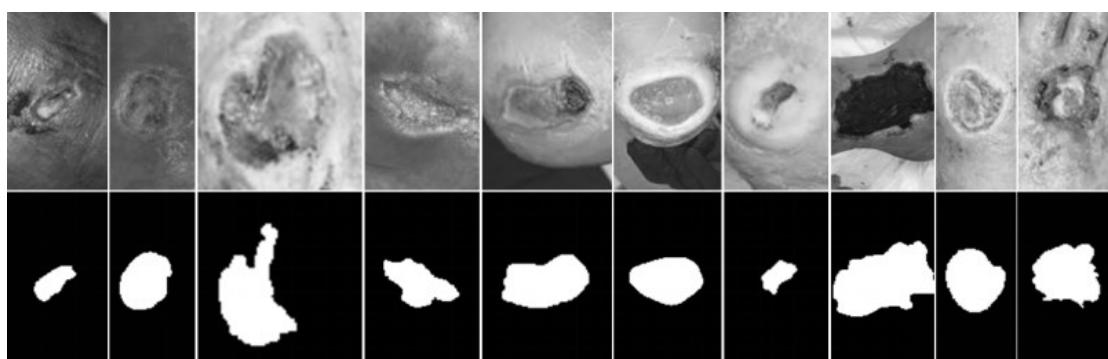
งานวิจัยที่เกี่ยวข้องกับการ deep learning โดยใช้ U-Net model โดย train กับจำนวนรูปผลใน dataset โดยผลลัพธ์จากการทดลองของ model จะถูกนำมาเทียบ PWAT score กับผลลัพธ์การประเมินโดยผู้เชี่ยวชาญทางการแพทย์และบ่งชี้การเรียนรู้ของ model ในครั้งต่อไป



รูปที่ 2.29 รูปภาพกระบวนการทำงานของ Deepskin Model

### 2.3.2 Fully automatic wound segmentation with deep convolutional neural networks

งานวิจัยเกี่ยวกับการใช้งาน deep convolutional neural network โดยให้มีผู้เชี่ยวชาญทางการแพทย์เป็นผู้กำหนดขอบเขตของแผลจาก dataset ภาพแพลงและนำไปใช้ train model โดยใช้เบริ่งเทียบเมทริกซ์ที่ได้จากโมเดลต่างชนิดกัน



รูปที่ 2.30 รูปภาพตัวอย่างของ wound-segmentation model

### 2.3.2.1 ชุดข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูลแพลตฟอร์มนี้ถูกรวบรวมตลอดสองปีจากผู้ป่วย 889 คน ที่ศูนย์บำบัดแพล และหลอดเดือด AZH มีภาพแพลที่เท้าจำนวน 1,109 ภาพ โดยถ่ายด้วยกล้อง Canon และ iPad Pro ภายใต้สภาพแสงและพื้นหลังที่หลากหลาย ก่อนจะนำไปใช้กับโมเดลลีกภาพทุกภาพถูกทำให้มีขนาดเท่ากัน (224x224 พิกเซล) ด้วยการครอบและเสริมช่องว่าง (zero-padding) โดยใช้ YOLOv3 ที่ฝึกบนชุดข้อมูลขนาด 1,010 ภาพที่ผ่านการระบุตำแหน่งแพลงแล้ว ผลการประเมินความแม่นยำอยู่ที่ค่า mAP เท่ากับ 0.939

### 2.3.2.2 รูปแบบการทดลอง

โครงสร้างการทดลองของโปรเจกต์นี้เริ่มจากการเก็บรวบรวมชุดข้อมูลภาพแพล เรื่อรังของผู้ป่วย จากนั้นนำภาพเหล่านี้มาทำการระบุตำแหน่งแพลงด้วยโมเดล YOLOv3 ที่พัฒนาขึ้น เพื่อใช้ในการครอบบริเวณแพลงและปรับขนาดภาพให้เท่ากัน (224x224 พิกเซล) ผ่านการใช้ padding เพื่อความสม่ำเสมอของขนาด หลังจากนั้นจึงนำภาพที่ผ่านการปรับแต่งมาใช้กับโมเดลการเรียนรู้ เชิงลึกสำหรับการแบ่งส่วนแพลง

### 2.3.2.3 ผลการทดลอง

ผลการทดลองแสดงให้เห็นว่าโมเดล MobileNetV2 + CCL มีประสิทธิภาพสูงใน การแบ่งส่วนแพลง โดยการประเมินผลใช้เมทริกซ์ที่สำคัญ ได้แก่ Precision, Recall, และ Dice score ซึ่งบ่งชี้ถึงความสามารถของโมเดลในการแยกพิกเซลที่เกี่ยวข้องกับแพลงอย่างแม่นยำ โมเดลนี้มีค่าเฉลี่ยของ Dice score สูงสุด แสดงถึงความแม่นยำที่เหนือกว่าในเชิงการประเมินเชิงปริมาณ สำหรับการแบ่งส่วนภาพแพลง

**ตารางที่ 2.2 ตารางเปรียบเทียบเมตริกซ์ของผลลัพธ์ที่ได้จากโมเดลแต่ละตัวในงานวิจัย Fully automatic wound segmentation with deep convolutional neural networks**

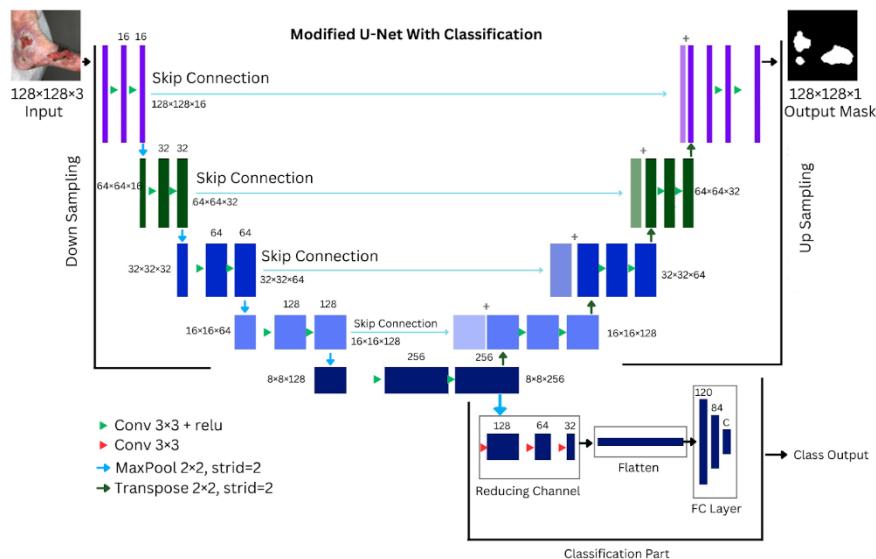
<b>Model</b>	<b>VGG16 (%)</b>	<b>SegNet (%)</b>	<b>U-Net (%)</b>	<b>Mask-RCNN (%)</b>	<b>MobileNetV2 (%)</b>	<b>MobileNetV2 + CCL (%)</b>
Precision	83.91	83.66	89.04	94.30	90.86	91.01
Recall	78.35	86.49	91.29	86.40	89.76	89.97
Dice	81.03	85.05	90.15	90.20	90.30	90.47

### 2.3.2.3 สรุปผลการทดลอง

ในงานวิจัยนี้มีเป้าหมายในการแบ่งส่วนภาพแพลงเรื่อรังโดยใช้โมเดล deep learning อายุ่ง MobileNetV2 ที่รวม CCL และทดสอบประสิทธิภาพเมื่อเทียบกับ SegNet VGG16 U-Net และ Mask-RCNN โมเดลนี้มีข้อดีคือความสามารถในการแยกพิกเซลได้แม่นยำและใช้งานกับชุดข้อมูลอื่นได้ อายุ่งไร้กีตาม ข้อจำกัดหลักคือจำนวนข้อมูลที่มีจำกัด ซึ่งอาจลดลงความแม่นยำของการทำนาย และการใช้โมเดลที่ซับซ้อนอาจทำให้ประสิทธิภาพการประมวลผลบนอุปกรณ์พกพาอย่างไม่เพียงพอ

### 2.3.3 Deep Learning for Automated Wound Classification and Segmentation

งานวิจัยนำเสนอการใช้โมเดล deep learning เช่น UNet และ EfficientNet ในการจำแนกประเภทและแยกส่วนภาพบาดแผล โดยมุ่งเน้นการปรับปรุงความแม่นยำในการระบุประเภทและตำแหน่งของบาดแผล งานวิจัยแสดงให้เห็นว่าเทคโนโลยีสามารถช่วยเพิ่มความรวดเร็วและความน่าเชื่อถือในการประเมินบาดแผล ซึ่งมีศักยภาพสูงในการสนับสนุนบุคลากรทางการแพทย์และยกระดับการดูแลผู้ป่วย



รูปที่ 2.31 รูปสถาปัตยกรรม Modified U-Net ที่ใช้ในงานวิจัย Deep Learning for Automated Wound Classification and Segmentation

#### 2.3.3.1 ชุดข้อมูลที่ใช้ในการทดลอง

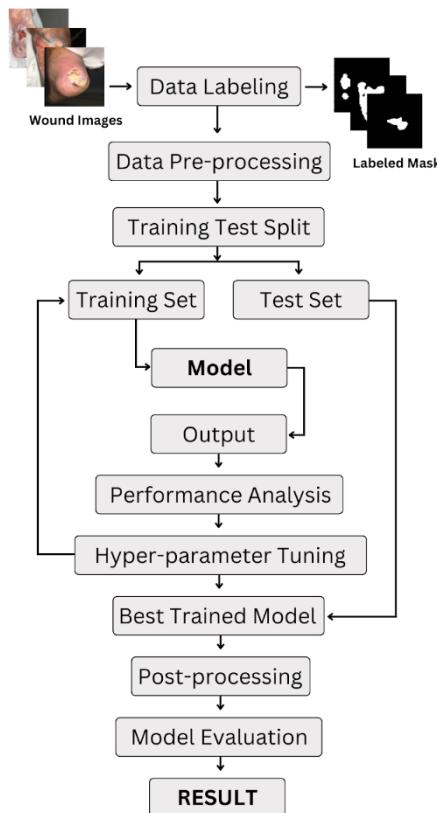
การทดลองนี้ใช้ชุดข้อมูลที่ครอบคลุมบาดแผล 4 ประเภท ได้แก่ แพลติดเชื้อ แพลงท์เท้า แพลงท์ขา และแพลงคดหัน ซึ่งคงข้อมูลจากชุดข้อมูล Medetech และภาพที่ได้จากการถ่ายภาพทางอินเทอร์เน็ต ข้อมูลทั้งหมดได้รับการปรับขนาดเป็น  $128 \times 128$  พิกเซล เพื่อให้สอดคล้องกับสำหรับการประมวลผลแบบจำลอง ภาพทุกภาพผ่านการทำเครื่องหมายอย่างแม่นยำโดยใช้เครื่องมือ Make Sense AI เพื่อสร้างมาสก์ที่แบ่งส่วนภาพซึ่งใช้ในการฝึกอบรมและประเมินผลการเรียนรู้ของแบบจำลอง

ตารางที่ 2.3 ตารางเปรียบเทียบปริมาณรูปภาพแต่ละประเภท

Type	Image count
Infected Wound	51
Foot Ulcer	52
Leg Ulcer	63
Pressure Ulcer	90

#### 2.3.3.2 รูปแบบการทดลอง

การทดลองในงานวิจัยนี้ประกอบด้วยกระบวนการหารายชื่อต้นเริ่มต้นจากการทำเครื่องหมายข้อมูล (Data Labeling) เพื่อสร้างมาสก์แบ่งส่วน จากนั้นทำการปรับแต่งข้อมูล (Data Pre-processing) และแบ่งข้อมูลออกเป็นชุดฝึกอบรมและชุดทดสอบ แบบจำลองจะถูกฝึกด้วยชุดฝึกอบรมและประเมินผลด้วยชุดทดสอบ โดยผ่านขั้นตอนการวิเคราะห์ผล (Performance Analysis) และการปรับค่าพารามิเตอร์ (Hyper-parameter Tuning) เพื่อเลือกแบบจำลองที่ดีที่สุด ก่อนทำการประเมินผลขั้นสุดท้ายและแสดงผลลัพธ์



**รูปที่ 2.32 รูปกระบวนการทดลองในงานวิจัย Deep Learning for Automated Wound Classification and Segmentation**

### 2.3.3.3 ผลการทดลอง

ผลการทดลองแสดงให้เห็นว่าแบบจำลองมีประสิทธิภาพสูงทั้งในการจำแนกประเภทและการแบ่งส่วนบาดแผล โดยในงานจำแนกประเภทมีค่าความถูกต้อง (Accuracy) 0.915 และ F1 Score 0.907 ซึ่งสะท้อนถึงความสามารถในการจัดหมวดหมู่บาดแผล ได้อย่างดี สำหรับงานแบ่งส่วน แบบจำลองมีความแม่นยำ (Precision) 0.947 และ Dice Score 0.931 พร้อมด้วย F1 Score 0.929 แสดงให้เห็นถึงความสามารถในการระบุตำแหน่งของบาดแผล ได้อย่างแม่นยำและมีความสอดคล้องกับข้อมูลจริง

## ตารางที่ 2.4 ตารางเปรียบเทียบเมตริกซ์ของผลลัพธ์ที่ได้จากโมเดลแต่ละตัวในงานวิจัย Deep Learning for Automated Wound Classification and Segmentation

Evaluation Metrics	Results
Dice Score	0.931
Precision	0.947
Recall	0.865
F1-Score	0.929

### 2.3.3.4 สรุปผลการทดลอง

สรุปข้อจำกัดของการศึกษานี้พบว่า แม้โมเดลจะให้ผลลัพธ์ที่ดีทั้งด้านการจำแนก (accuracy 0.915, F1-score 0.907) และการแบ่งส่วนภาพ (precision 0.947, Dice score 0.931, F1-score 0.929) แต่ยังมีข้อจำกัดที่สำคัญคือ ชุดข้อมูลฝึกสอนยังขาดความหลากหลายที่จะช่วยให้โมเดลทำงานได้กับbacgroundที่มีลักษณะแตกต่างกัน ทำให้ต้องมีการขยายชุดข้อมูลและปรับปรุงด้วยเทคนิค data augmentation ขึ้นสูง นอกจากนี้ โมเดลยังอยู่ในขั้นการวิจัย ยังไม่พร้อมสำหรับการใช้งานในคลินิกหรือที่บ้าน จำเป็นต้องพัฒนาเป็นแอปพลิเคชันมือถือเพื่อให้สามารถนำไปใช้งานได้จริง รวมถึงยังไม่สามารถเข้าถึงการใช้งานในพื้นที่ห่างไกลหรือพื้นที่ขาดแคลนได้ในปัจจุบัน ซึ่งเป็นประเด็นที่ต้องได้รับการพัฒนาต่อไป

## 2.4 แอปพลิเคชันที่เกี่ยวข้อง

### 2.4.1 ImitoMeasure

ImitoMeasure - แอปพลิเคชันวัดขนาดแผล โดยในตัวแอปจะมีการถ่ายภาพแผลเพื่อวัดขนาดและวัดความรุนแรงของแผล โดยมีการให้คะแนน

ตารางที่ 2.1 ตารางเปรียบเทียบ Function ของแอปพลิเคชัน ImitoMeasure

กิจกรรม	Function ของ ImitoMeasure	Function ที่จะนำมาประยุกต์และปรับใช้
การถ่ายรูปแพล	หน้าถ่ายรูปเพื่อทำการถ่ายแพล ต้องวางสิ่งของหรือ object ที่เราขึ้นมาในชีวิตจริงเพื่อเป็นสิ่งอ้างอิงขนาดได้ เพื่อใส่ขนาดของวัตถุอ้างอิงแล้วจึงกำหนดวงแพลตามต้องการ	การถ่ายรูปแสดงผลโดยไม่ต้องมี object เพื่ออ้างอิงขนาด และมีการเก็บข้อมูลแพลของผู้ป่วยไว้เป็น report
การบันทึก ตำแหน่งแพล	มีการบันทึกแพลว่าอยู่บริเวณส่วนไหนของร่างกาย	มีการเก็บตำแหน่งของแพลเป็นหมวดหมู่ในส่วนของผู้ดูแล หมอและพยาบาล
การแสดงข้อมูลแพล	แสดงผลขนาดแพล เส้นรอบวงแพล ความลึกของแพล และวันที่วัดแพลซึ่งสามารถคุปร่วติข้อมูลหลังได้	แสดงชนิด ความรุนแรง และวิธีการคุณภาพแพลเบื้องต้นแสดงการแนะนำการคุณภาพจาก การแนะนำของหมอ และพยาบาลผู้เชี่ยวชาญ
การเก็บข้อมูลการรักษา	เก็บ report ของผู้ป่วยและรายละเอียดแพล	มีการเก็บข้อมูลการรักษา และข้อมูลการแนะนำการรักษาจากหมอ พยาบาล
การติดตามการรักษา	ติดตามผลของการรักษา และ รายละเอียดของแพลแต่ละครั้งที่ถ่าย	มีผู้ดูแล พยาบาล และหมอติดตามอาการของแพลงกายในแอป

## บทที่ 3

### การออกแบบ

#### 3.1 ภาพรวมของระบบ

Mobile application สำหรับจัดการคูณผู้ป่วยแพลกต์ทับซึ่งประมวลผลรูปภาพโดย model deep learning โปรเจกต์นี้มีการออกแบบซอฟต์แวร์โดยแบ่งออกเป็นสองส่วนหลัก คือฝั่ง Client-Side และฝั่ง Server-Side โดยระบบนี้จะถูกออกแบบในลักษณะที่ฝั่ง Client จะทำหน้าที่แสดงผลข้อมูลและรับอินพุตจากผู้ใช้ ขณะที่ฝั่ง Server จะรับผิดชอบในการจัดการข้อมูลส่วนหลังบ้านอย่าง ประมวลผลภาพ และการจัดเก็บข้อมูล

##### 3.1.1 การออกแบบ Client-Side

Client-Side ประกอบด้วย 2 ส่วนหลัก ได้แก่ แอปพลิเคชันมือถือและระบบ ERP ที่พัฒนาโดยใช้เทคโนโลยีต่างกัน แต่มีการเชื่อมต่อผ่าน API ของฝั่งเซิร์ฟเวอร์เพื่อประมวลผลข้อมูลที่เกี่ยวข้อง โดยมีรายละเอียดดังนี้

###### 3.1.1.1 แอปพลิเคชันบนมือถือ (Flutter)

แอปพลิเคชันมือถือถูกพัฒนาด้วย Flutter เป็นระบบหลักซึ่งมีฟังก์ชันหลักของแอปนี้คือการแสดงผลและจัดการอินพุตของผู้ใช้รวมไปถึงการส่งรูปภาพแพลไประมวลผลที่ฝั่งหลังบ้าน นอกจากนี้ยังมีฟังก์ชันในการติดต่อสื่อสารกันของผู้ใช้แต่ละบบทบาท

###### 3.1.1.2 ระบบ ERP (Angular)

ถูกพัฒนาด้วย Angular โดยระบบ ERP จะใช้สำหรับการจัดการข้อมูลผู้ใช้และแสดง Dashboard โดยเป็นระบบที่ผูกคูณระบบสามารถจัดการข้อมูลต่างในระบบได้

##### 3.1.2 การออกแบบสถาปัตยกรรมฝั่ง Server-Side

Server-Side ประกอบด้วยส่วนประกอบต่างๆ ที่รับผิดชอบการประมวลผลภาพ การจัดการ API และการเชื่อมต่อกับฐานข้อมูล ซึ่งมีรายละเอียดดังนี้

###### 3.1.2.1 Backend (NestJS)

พัฒนาใช้ NestJS สามารถจัดการ API ไมเดลเอนทิตี้ต่าง ๆ รวมถึงการ การจัดการเส้นทาง (Routing) และการทำ Caching เพื่อเพิ่มประสิทธิภาพในการประมวลผลข้อมูล

###### 3.1.2.2 Backend (FastAPI)

พัฒนาใช้ FastAPI เป็นตัวกลางในการเชื่อมต่อระหว่าง Backend กับไมเดลทำหน้าที่ในการรับส่งอินพุตจากตัว Backend เป้าสู่ส่วน Model และส่งค่าที่ predict ได้จากไมเดลกลับสู่ Backend

### 3.1.2.3 Model การประมวลผลรูปภาพ

ระบบนี้มีโมเดลสองประเภทในการประมวลผลรูปภาพ คือ

#### 3.1.2.3.1 Wound Segmentation

ใช้สำหรับการทำ segmentation บาดแผล โดยจะทำหน้าที่แบ่งแยกส่วนของภาพที่เป็นบาดแผลออกจากพื้นหลัง

#### 3.1.2.3.1 Wound Classification

ใช้สำหรับการจำแนกประเภทบาดแผล เพื่อช่วยวินิจฉัยประเภทบาดแผล และส่งผลลัพธ์กลับสู่ Backend

### 3.1.2.4 ฐานข้อมูล (PostgreSQL)

สำหรับการจัดการข้อมูลทั้งหมดของระบบ เช่น ข้อมูลผู้ใช้และข้อมูลบาดแผล ใช้ PostgreSQL เป็นฐานข้อมูลแบบ relational database

### 3.1.2.5 Containerization (Docker)

Docker เป็นแพลตฟอร์ม Containerization ที่ช่วยบรรจุแอปพลิเคชันและทุกสิ่งที่จำเป็นสำหรับการทำงานลงในคอนเทนเนอร์ (Container) ซึ่งเป็นหน่วยแยกอิสระที่สามารถรันได้บนทุกสภาพแวดล้อม โดยไม่ต้องกังวลเรื่องความแตกต่างของระบบปฏิบัติการหรือการตั้งค่าเซิร์ฟเวอร์

### 3.1.2.6 Kubernetes (K8s)

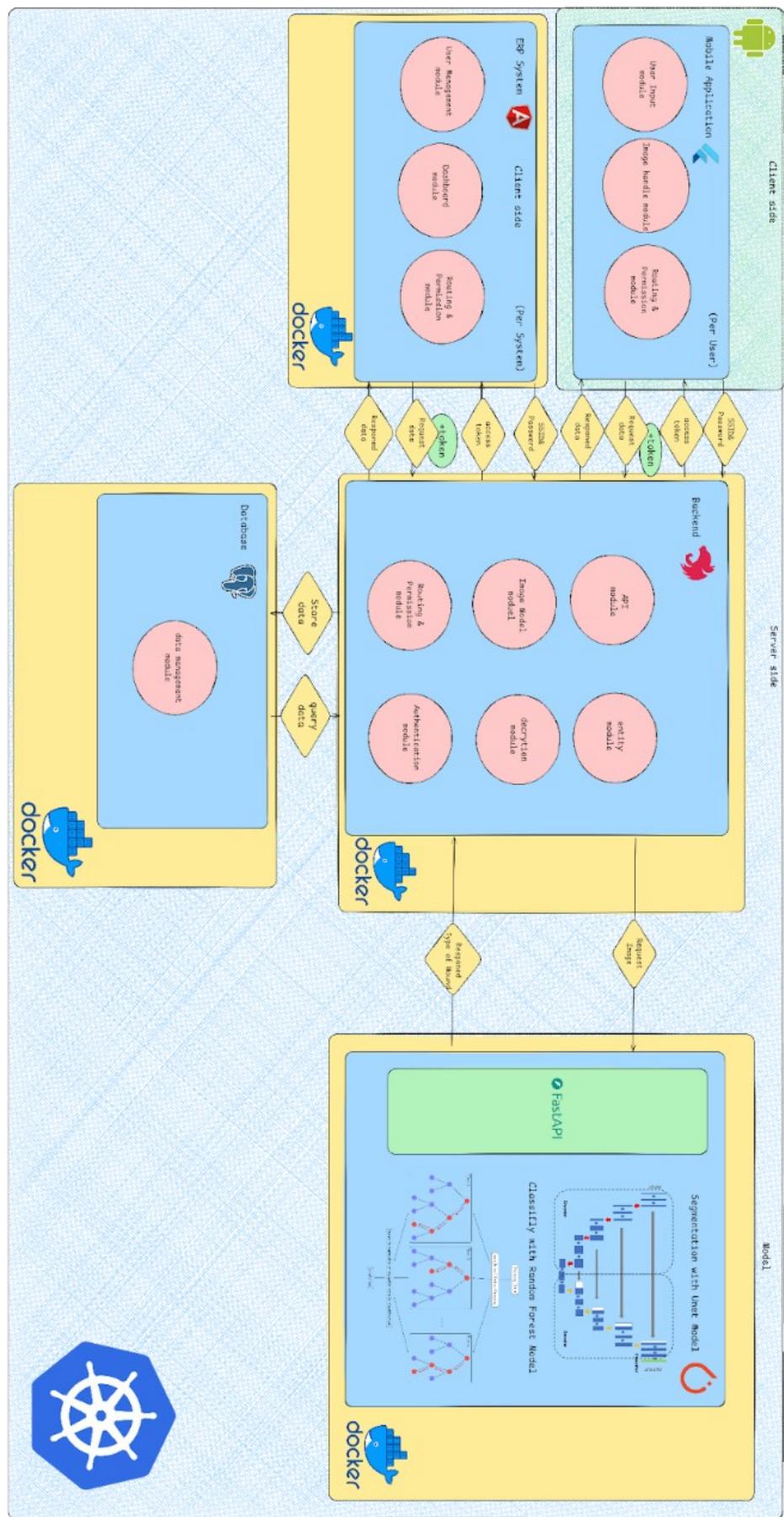
Kubernetes เป็น Container Orchestrator ที่ช่วยจัดการ, ปรับขนาด และควบคุมชีวิตของ Containers โดยทำงานร่วมกับ Container Runtime เช่น Docker Container

### 3.1.2.7 Cloudflare Tunnel

Cloudflare Tunnel ใช้เชื่อมต่อเซิร์ฟเวอร์หรือแอปพลิเคชันภายใน (เช่น Kubernetes Cluster, Backend APIs หรือ Localhost) เข้ากับอินเทอร์เน็ต

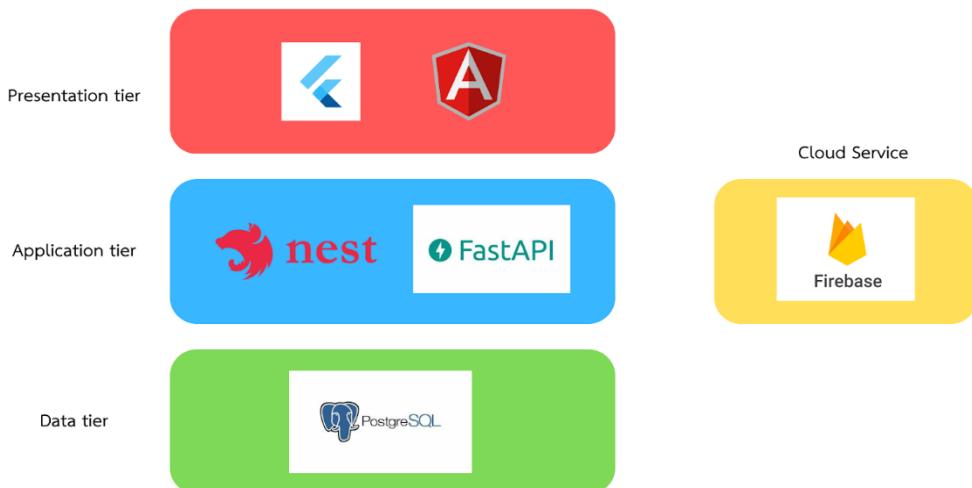
### 3.1.3 การเชื่อมต่อและการทำงานร่วมกันของระบบ

การทำงานร่วมกันระหว่าง Client-Side และ Server-Side ถูกเชื่อมต่อผ่าน API ที่จัดการโดย Backend ซึ่งจะทำหน้าที่ในการรับส่งข้อมูลและรูปภาพจากแอปมือถือและระบบ ERP ไปยัง Server เพื่อประมวลผลและตอบกลับข้อมูลที่ผ่านการวิเคราะห์แล้ว ทั้งนี้การเชื่อมต่อดังกล่าวจะถูกป้องกันโดยระบบยืนยันตัวตน (Authentication) และการกำหนดสิทธิ์การเข้าถึงข้อมูล



รูปที่ 3.1 ภาพการออกแบบของระบบซอฟต์แวร์โดยรวม

### 3.2 การออกแบบสถาปัตยกรรมระบบซอฟต์แวร์



รูปที่ 3.2 ภาพการออกแบบสถาปัตยกรรมระบบซอฟต์แวร์ในรูปแบบ 3-Tier

จากรูป 3.2 จะเป็นการออกแบบสถาปัตยกรรมระบบซอฟต์แวร์โดยใช้แนวคิด 3-Tier Architecture ซึ่งเป็นรูปแบบสถาปัตยกรรมที่มีการแยกหน้าที่ของระบบออกเป็นสามชั้นหลัก ได้แก่ ชั้นนำเสนองาน (Presentation Tier) ชั้นการประมวลผล (Application Tier) ชั้นจัดเก็บข้อมูล (Data Tier) และส่วนของบริการคลาวด์ (Cloud Service) แต่ละชั้นมีหน้าที่และบทบาทที่ชัดเจนในการทำงาน

#### 3.2.1 Presentation Tier

ชั้นที่เกี่ยวข้องกับการแสดงผลและการรับข้อมูลจากผู้ใช้ ในระบบนี้มีสองส่วนหลักคือ แอปพลิเคชันมือถือซึ่งพัฒนาโดย Flutter และระบบ ERP ซึ่งพัฒนาโดย Angular

#### 3.2.2 Application Tier

ชั้นที่รับผิดชอบการประมวลผลเชิงตรรกะและจัดการข้อมูล ซึ่งทำหน้าที่เป็นตัวกลางระหว่างชั้น Presentation และ Data ชั้นนี้ทำงานผ่าน Backend ที่พัฒนาด้วย NestJS และโอมเดลการประมวลผลรูปภาพ

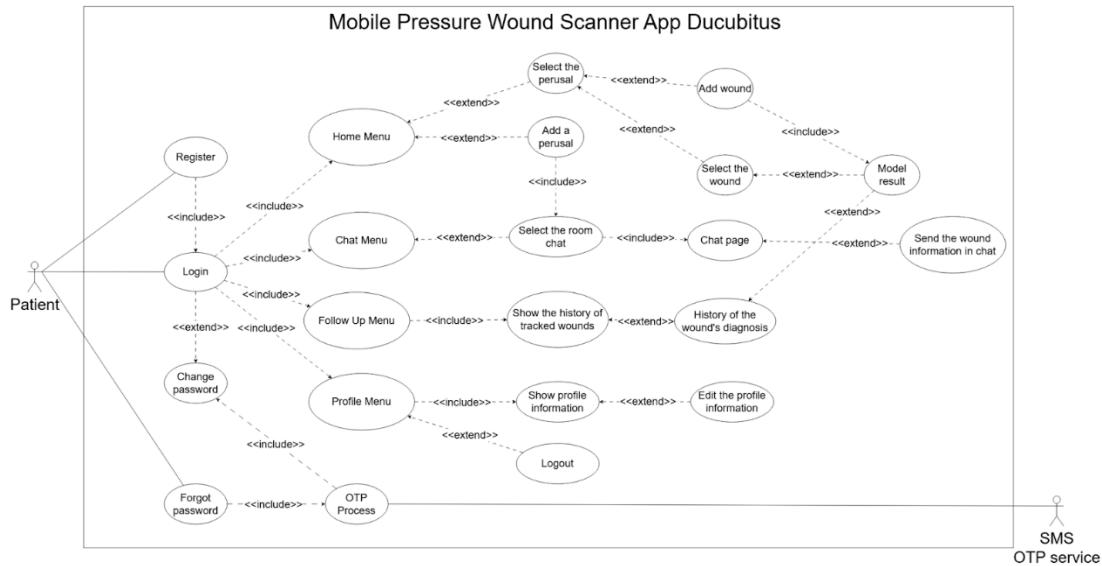
#### 3.2.3 Data Tier

ชั้นที่เกี่ยวข้องกับการจัดเก็บและจัดการข้อมูลอย่างถาวร ในระบบนี้ใช้ฐานข้อมูล PostgreSQL ระบบแคช (Redis)

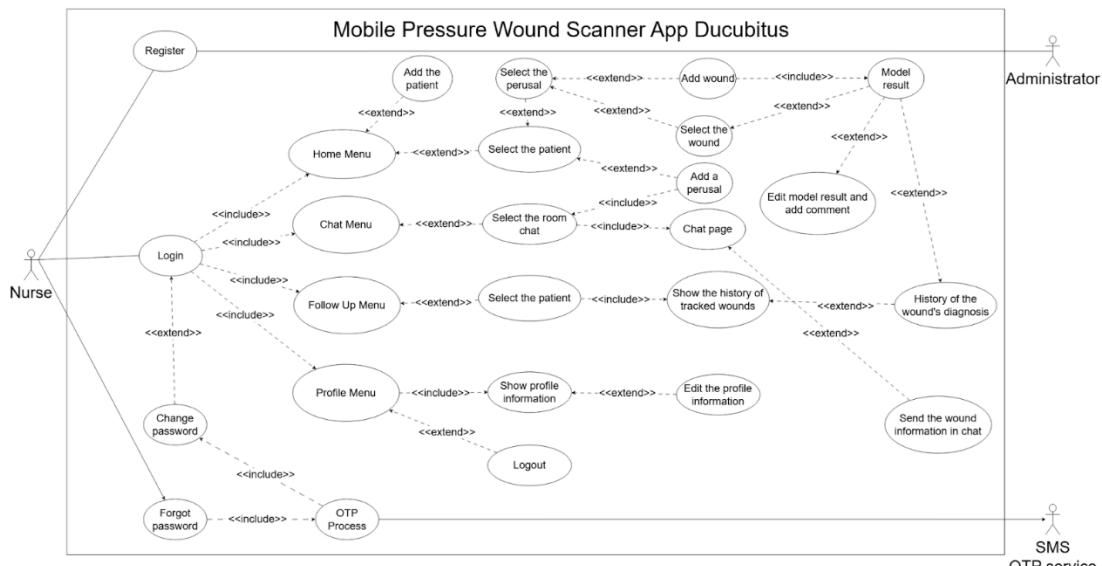
#### 3.2.4 Cloud Service

ส่วนของบริการ Cloud ซึ่งในที่นี้จะใช้บริการของ Firebase ในการทำระบบแชทแบบ Realtime

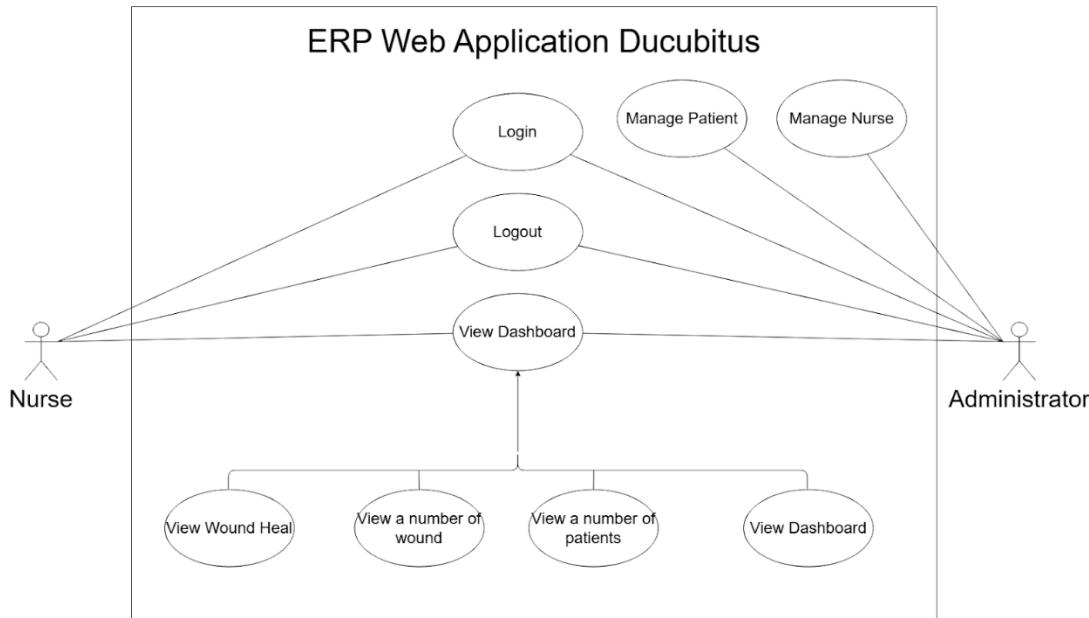
### 3.3 การออกแบบ Use Case Diagram



รูปที่ 3.3 Use Case Diagram ของผู้病



รูปที่ 3.4 Use Case Diagram ของผู้ดูแล



รูปที่ 3.5 Use Case Diagram ของ ERP System

จากรูปที่ 3.3 และ 3.4 ในระบบมี Actor จำนวน 3 ประเภท ได้แก่

### 3.3.1 Patient (ผู้ป่วย)

ผู้ป่วยเป็นผู้ใช้งานระบบที่สามารถดำเนินการต่างๆ ได้ดังต่อไปนี้

- 1) ลงทะเบียน (Register) และเข้าสู่ระบบ (Login) เพื่อใช้งานระบบ
- 2) เพิ่มรายการตรวจ รวมถึงเพิ่มรายการแพลที่ต้องการตรวจสอบตรวจแต่ละครั้ง สามารถดูรายละเอียดของแพลที่ตรวจได้
- 3) เพิ่มรายการตรวจ รวมถึงเพิ่มรายการแพลที่ต้องการตรวจสอบตรวจแต่ละครั้ง สามารถดูรายละเอียดของแพลที่ตรวจได้
- 4) ติดต่อและพูดคุยกับพยาบาลเพื่อให้ข้อมูลเพิ่มเติมเกี่ยวกับแพลที่ได้รับการตรวจ
- 5) ติดตามความคืบหน้าของแพลที่อยู่ระหว่างการรักษาหรือที่ได้รับการรักษาเสร็จสิ้น แล้ว
- 6) แก้ไขข้อมูลส่วนตัวของตนเอง
- 7) เปลี่ยนรหัสผ่าน โดยใช้ OTP (One-Time Password) เพื่อยืนยันตัวตนก่อนดำเนินการเปลี่ยนรหัสผ่าน

### 3.3.2 Nurse (พยาบาล)

พยาบาลเป็นบุคลากรทางการแพทย์ที่สามารถดำเนินการต่างๆ ได้ดังนี้

- 1) ลงทะเบียน (Register) และเข้าสู่ระบบ (Login) เพื่อใช้งานระบบ
- 2) ใช้งานระบบแพทย์เพื่อพูดคุยและให้คำแนะนำเกี่ยวกับการดูแลแพลแก่ผู้ป่วยที่อยู่ใน การดูแล

- 3) บันทึกข้อมูลแพลและปรับแก้ไขรายละเอียดแพลของผู้ป่วยที่อยู่ภายใต้การดูแล
- 4) เพิ่มรายการตรวจ รวมถึงเพิ่มรายการแพลของผู้ป่วยที่ต้องการตรวจสอบ
- 5) ดูรายละเอียดการเข้าตรวจของผู้ป่วยรวมถึงผลการประมวลผลของโมเดลที่ระบุระดับของแพลก์ทับและแนวทางการรักษาเบื้องต้น
- 6) ติดตามความคืบหน้าเกี่ยวกับแพลของผู้ป่วยที่อยู่ภายใต้การดูแล

### 3.3.3 Administrator

Administrator เป็นผู้รับผิดชอบในการทำงานของระบบ โดยมีลิทธิ์ในการดำเนินการดังต่อไปนี้

- 1) เข้าถึงข้อมูล Dashboard ที่แสดงสถิติเกี่ยวกับระบบ เช่น จำนวนผู้ใช้งาน จำนวนพยาบาล จำนวนผู้ป่วยที่กำลังรับการรักษา จำนวนผู้ป่วยที่รักษาสำเร็จ ข้อมูลแพล และข้อมูลอื่นๆ ที่เกี่ยวข้อง
- 2) เข้าถึงรายละเอียดข้อมูลของผู้ป่วย พยาบาล และแพทย์ รวมถึงลิทธิ์ในการปรับปรุงแก้ไข เพิ่ม หรือ ลบข้อมูลของผู้ใช้งานตามความจำเป็น

ตารางที่ 3.1 Register Use Case

Use Case:	Register (ลงทะเบียน)
Actor:	Patient
Pre-Condition:	Actor ปัจจุบันไม่มีบัญชีในระบบ
Post-Condition:	Actor ลงทะเบียนสำเร็จและสามารถเข้าสู่ระบบได้
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor เปิดหน้า สร้างบัญชีผู้ใช้งาน</li> <li>2. Actor กรอกข้อมูลที่จำเป็นดังนี้ <ul style="list-style-type: none"> <li>• เลขบัตรประชาชน (ต้องเป็นตัวเลข 13 หลักเท่านั้น)</li> <li>• ชื่อจริง นามสกุล</li> <li>• เลือกเพศ (ชาย / หญิง)</li> <li>• เบอร์โทรศัพท์ (ต้องเป็นตัวเลข 10 หลัก)</li> <li>• วันเกิด (รูปแบบ: ปี-เดือน-วัน หรือ ปปปป-คค-วว)</li> <li>• รหัสผ่าน (ต้องมีความยาวขั้นต่ำ 6 ตัวอักษร)</li> <li>• ยืนยันรหัสผ่าน (ต้องตรงกับรหัสผ่านที่กรอกไว้)</li> </ul> </li> <li>3. Actor กดปุ่ม "ลงทะเบียน"</li> </ol>

	<p>4. ระบบทำการตรวจสอบข้อมูลที่กรอก</p> <ul style="list-style-type: none"> <li>ถ้าข้อมูลถูกต้อง ระบบจะบันทึกข้อมูลและแจ้งว่า ลงทะเบียนสำเร็จ</li> <li>ถ้าข้อมูลไม่ถูกต้อง ระบบจะแสดงข้อความแจ้งเตือนเพื่อให้ Actor แก้ไขข้อมูล</li> </ul> <p>5. หากลงทะเบียนสำเร็จ Actor สามารถใช้ตัวเลขบัตรประชาชนและรหัสผ่านที่ลงทะเบียนไว้เพื่อ Login เข้าใช้งานระบบได้</p>
AlternativeFlow:	<ul style="list-style-type: none"> <li>- หาก Actor กรอกข้อมูลไม่ครบ ระบบจะแสดงข้อความแจ้งเตือน จนกว่าจะได้รับการแก้ไขให้ครบ</li> <li>- หากเบอร์โทรศัพท์หรือเลขบัตรประชาชน ซ้ำกับที่มีอยู่ในระบบ ระบบจะแจ้งเตือนว่า "ข้อมูลนี้ถูกใช้ไปแล้ว"</li> </ul>
Exception Flow:	หาก Actor ตัดสินใจไม่ลงทะเบียน สามารถปิดหน้าจอได้ หรือกด ข้อนกลับ เพื่อกลับไปหน้า Login

ตารางที่ 3.2 Login Use Case

Use Case:	Login (เข้าสู่ระบบ)
Actor:	Patient, Nurse, Admin
Pre-Condition:	Actor ยังไม่ได้ Login เข้าสู่ระบบ Actor มีบัญชีที่ลงทะเบียนแล้วในระบบ
Post-Condition:	Actor เข้าสู่ระบบสำเร็จและสามารถใช้งานระบบได้
Main Flow:	<ol style="list-style-type: none"> <li>Actor เปิดหน้า เข้าสู่ระบบ</li> <li>Actor กรอกข้อมูล <ul style="list-style-type: none"> <li>เลขบัตรประชาชน (ต้องเป็นตัวเลข 13 หลักเท่านั้น)</li> <li>รหัสผ่าน (ต้องถูกต้องตามที่ตั้งไว้)</li> </ul> </li> <li>Actor กดปุ่ม "Login" ระบบทำการตรวจสอบข้อมูล <ul style="list-style-type: none"> <li>หากข้อมูลถูกต้อง ระบบอนุญาตให้เข้าสู่ระบบและนำไปยังหน้า Home</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>หากข้อมูลไม่ถูกต้อง ระบบแสดงข้อความแจ้งเตือนและให้กรอกข้อมูลใหม่</li> </ul>
AlternativeFlow:	หาก Actor กรอก เลขบัตรประชาชนหรือรหัสผ่านผิด ระบบจะแสดงข้อความแจ้งเตือนว่า "เลขบัตรประชาชนหรือรหัสผ่านไม่ถูกต้อง กรุณาลองใหม่"
Exception Flow:	หาก Actor ตัดสินใจไม่เข้าสู่ระบบ สามารถกดปิดแอปพลิเคชันได้

### ตารางที่ 3.3 Review wound records information Use Case

Use Case:	Forgot Password (ลืมรหัสผ่าน)
Actor:	Patient, Nurse
Pre-Condition:	Actor มีบัญชีในระบบและเคลื่อนไหว
Post-Condition:	Actor สามารถตั้งค่ารหัสผ่านใหม่และใช้เข้าสู่ระบบได้
Main Flow:	<ol style="list-style-type: none"> <li>Actor กดปุ่ม "ลืมรหัสผ่าน" บนหน้า Login</li> <li>ระบบแสดงหน้ากรอกหมายเลขโทรศัพท์ ที่ใช้งานทะเบียน</li> <li>Actor กรอกข้อมูลและกดปุ่ม "ดำเนินการต่อ"</li> <li>ระบบตรวจสอบว่าหมายเลขโทรศัพท์ตรงกัน กับที่ลงทะเบียนไว้</li> <li>ระบบส่ง OTP (รหัสยืนยัน 6 หลัก) ไปยังหมายเลขโทรศัพท์ของ Actor</li> <li>Actor กรอก OTP และกดปุ่ม "ยืนยัน OTP"</li> <li>ระบบตรวจสอบความถูกต้องของ OTP <ul style="list-style-type: none"> <li>ถ้า OTP ถูกต้อง → ระบบให้ Actor ตั้งรหัสผ่านใหม่</li> <li>ถ้า OTP ผิด → ระบบแจ้งเตือนและให้ลองใหม่ (30 วินาที)</li> </ul> </li> <li>Actor กรอกรหัสผ่านใหม่ และยืนยันรหัสผ่านใหม่</li> <li>Actor กดปุ่ม "รีเซ็ตรหัสผ่าน" และจะถูกส่งกลับไปที่หน้าเข้าสู่ระบบ</li> </ol>
AlternativeFlow:	หาก OTP หมดอายุ (30 วินาทีผ่านไป) → ปุ่มส่ง OTP ใหม่จะ

	แสดงให้กด
Exception Flow:	Actor ไม่ได้รับ OTP → กด "ส่งรหัสใหม่" หาก Actor ใช้หมายเลขโทรศัพท์ผิด → ระบบแจ้งเตือนว่า "เกิดข้อผิดพลาดในการส่ง OTP"

ตารางที่ 3.4 Select Perusal Use Case

Use Case:	Select Perusal (เลือกการตรวจ)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor (Nurse) ต้องมีผู้ป่วยในการดูแล
Post-Condition:	Actor (Patient) สามารถเลือกดูรายการการตรวจของตนเองได้ Actor (Nurse) สามารถเลือกดูรายการการตรวจของผู้ป่วยที่เกี่ยวข้องได้
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor เข้าสู่ระบบและไปที่ "Home Menu"             <ol style="list-style-type: none"> <li>1.1. Actor (Nurse) ต้องเลือกผู้ป่วยที่ต้องการดูประวัติการตรวจ ก่อน</li> <li>1.2. ระบบแสดงรายการ Perusal (ประวัติการตรวจ) ที่สามารถเข้าถึงได้</li> </ol> </li> <li>2. Actor เลือก Perusal ที่ต้องการ</li> <li>3. Actor สามารถดูรายละเอียด หรือเพิ่มข้อมูลเพิ่มเติมได้</li> </ol>
Alternative Flow:	ถ้ายังไม่มี Perusal ในระบบ → ระบบแจ้ง "กดปุ่มขวาล่างเพื่อเพิ่มรายการตรวจ"
Exception Flow:	-

### ตารางที่ 3.5 Add Perusal Use Case

Use Case:	Add Perusal (เพิ่มการตรวจใหม่)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor (Nurse) ต้องมีผู้ป่วยในการดูแล
Post-Condition:	Actor (Patient) สามารถเลือกคุณรายการตรวจของตนได้ Actor (Nurse) สามารถดูรายการตรวจผู้ป่วยได้
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor เข้าสู่ระบบและ ไปที่ "Home Menu"             <ol style="list-style-type: none"> <li>1.1. Actor (Nurse) ต้องเลือกผู้ป่วยที่ต้องการเพิ่มการตรวจก่อน</li> </ol> </li> <li>2. Actor กดเลือกเครื่องหมาย “+” ขวาน้ำด้านซ้ายของ Add Perusal</li> <li>3. ระบบแสดง Dialog Pop Up ให้ Actor ยืนยันเพื่อสร้างการตรวจ</li> <li>4. Actor กด "ยืนยัน"</li> <li>5. ระบบบันทึกข้อมูลลงฐานข้อมูล</li> </ol>
AlternativeFlow:	-
Exception Flow:	Actor กดยกเลิกบน Dialog Pop Up

### ตารางที่ 3.6 Select Wound Use Case

Use Case:	Select Wound (เลือกแผล)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor (Patient) ต้องมีประวัติการตรวจมาก่อน Actor (Nurse) ต้องมีผู้ป่วยในการดูแล
Post-Condition:	Actor สามารถดูรายละเอียดแผลที่เลือกได้
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor ไปที่ "Home Menu"             <ol style="list-style-type: none"> <li>1.1. Actor (Nurse) ต้องเลือกผู้ป่วยที่ต้องการดูรายละเอียดแผล ก่อน</li> </ol> </li> </ol>

	<p>2. Actor กดเลือกรายการตรวจที่ต้องการ</p> <p>3. ระบบแสดงรายการแพลที่เข้าไปคุ้มครองอยู่ในระบบ</p> <p>4. Actor เลือกแพลที่ต้องการคุ้มครองอยู่ในระบบ</p> <p>5. ระบบแสดงข้อมูลของแพลที่เลือก</p>
Alternative Flow:	หากไม่มีแพลในระบบ → ระบบแจ้งเตือน ""กดปุ่มขวาล่างเพื่อเพิ่มรายการแพล""
Exception Flow:	-

### ตารางที่ 3.7 Add Wound Use Case

Use Case:	Add Wound (เพิ่มแพลใหม่)
Actor:	Patient, Nurse
Pre-Condition:	<p>Actor ต้องเข้าสู่ระบบแล้ว</p> <p>Actor (Patient) ต้องมีประวัติการตรวจมา ก่อน</p> <p>Actor (Nurse) ต้องมีผู้ป่วยในการดูแล</p>
Post-Condition:	มีข้อมูลแพลใหม่ถูกบันทึกลงในระบบ
Main Flow:	<p>1. Actor เข้าสู่ระบบและไปที่ "Home Menu"</p> <p>    1.1 Actor (Nurse) ต้องเลือกผู้ป่วยที่ต้องการเพิ่มแพลก่อน</p> <p>2. Actor กดเลือกรายการตรวจที่ต้องการ</p> <p>3. Actor กดเลือกเครื่องหมาย "+" ขวาล่างของจอเพื่อ Add Wound</p> <p>4. Actor กรอกข้อมูลที่จำเป็น ดังนี้</p> <ul style="list-style-type: none"> <li>• อัปโหลด รูปภาพแพล</li> <li>• ระบุ วัยวะที่แพลเกิด</li> <li>• เลือกประเภทแพล (แพลใหม่ / แพลเก่า)</li> </ul> <p>        • หากเป็นแพลเก่า → ต้องเลือกแพลเดิมที่ต้องการอัปเดต</p> <p>5. Actor กด "ยืนยันข้อมูลเพื่อส่งประมวลผล"</p>

Alternative Flow:	หากกรอกข้อมูลไม่ครบถ้วน → ระบบจะไม่อนุญาตให้กดบันทึกข้อมูล
Exception Flow:	Actor สามารถกดย้อนกลับเพื่อออกจากหน้าการเพิ่มแพล็คได้

### ตารางที่ 3.8 Model Result Use Case

Use Case:	Model Result (ผลลัพธ์จากโภเมเดลประมวลผลแพล็ค)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor ต้อง Add Wound ในระบบแล้ว
Post-Condition:	ผลลัพธ์จากโภเมเดลถูกแสดงและบันทึกในระบบ
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor เข้าสู่ระบบและไปที่ "Home Menu"             <ol style="list-style-type: none"> <li>1.1 Actor (Nurse) ต้องเลือกผู้ป่วยที่ต้องการเพิ่มแพล็คก่อน</li> </ol> </li> <li>2. Actor กดเลือกรายการตรวจที่ต้องการ</li> <li>3. Actor กรอกข้อมูลเพื่อเพิ่มแพล็ค (Add Wound) และยืนยันการบันทึก</li> <li>4. ระบบประมวลผล แสดงผลลัพธ์ของแพล็ค และ รายละเอียดต่างๆ             <ul style="list-style-type: none"> <li>• วันที่บันทึกข้อมูล</li> <li>• ภาพถ่ายแพล็คที่บันทึกในระบบ</li> <li>• ระดับความรุนแรงของแพล็ค</li> <li>• คำอธิบายเกี่ยวกับระดับแพล็คและลักษณะของแพล็ค</li> <li>• คำแนะนำการรักษาเบื้องต้น</li> </ul> </li> </ol>
AlternativeFlow:	หากระบบไม่สามารถประมวลผลแพล็คได้ → ระบบแจ้งเตือนให้ Actor ลองใหม่หรือตรวจสอบข้อมูลรูปภาพแพล็ค
Exception Flow:	-

### ตารางที่ 3.9 Edit Model Result and Add Comment Use Case

Use Case:	Edit Model Result and Add Comment (แก้ไขผลลัพธ์จากโน้ตเดลและเพิ่มความคิดเห็น)
Actor:	Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor ต้องมีผู้ป่วยในการดูแล
Post-Condition:	ผลลัพธ์ของโน้ตเดลถูกแก้ไขหรือมีการเพิ่มความคิดเห็นสำเร็จ
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor เข้าสู่ระบบและไปที่ "Home Menu"             <ol style="list-style-type: none"> <li>1.1 Actor (Nurse) ต้องเลือกผู้ป่วยที่ต้องการแก้ไขผลลัพธ์แพลหรือเพิ่มความคิดเห็น</li> </ol> </li> <li>2. Actor กดเลือกรายการตรวจสอบที่ต้องการ</li> <li>3. Actor เลือกแพลที่ต้องการแก้ไขผลลัพธ์หรือเพิ่มความคิดเห็น</li> <li>4. ระบบแสดงผลลัพธ์ของแพลที่มีอยู่</li> <li>5. Actor กด Icon เพื่อแก้ไขข้อมูล</li> <li>5. Actor เลือกสามารถแก้ไขระดับความรุนแรงของแพล และเพิ่มความคิดเห็นได้</li> <li>6. Actor กด "รายงานผลการตรวจ"</li> <li>7. ระบบอัปเดตผลลัพธ์หรือบันทึกความคิดเห็นสำเร็จ</li> </ol>
AlternativeFlow:	-
Exception Flow:	หาก Actor ต้องการยกเลิกการแก้ไข → สามารถยกยื่นกลับได้

### ตารางที่ 3.10 Select the Room Chat Use Case

Use Case:	Select the Room Chat (เลือกห้องแชท)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor (Patient) ต้องสร้างการตรวจก่อน Actor (Nurse) ต้องแก้ไขผลลัพธ์จากโภเมเดลหรือเพิ่มความคิดเห็นก่อน
Post-Condition:	Actor สามารถเข้าสู่ห้องแชทและส่งข้อความได้
Main Flow:	1. Actor เข้าสู่ระบบและไปที่ "Chat Menu" 2. ระบบแสดงรายการ ห้องแชท ที่ Actor สามารถเข้าถึงได้ 3. Actor เลือกห้องแชทที่ต้องการ
AlternativeFlow:	หาก Actor ไม่มีห้องแชทในระบบ → ระบบแจ้งเตือนว่า "ไม่มีห้องแชท กรุณาสร้างการตรวจ"
Exception Flow:	-

### ตารางที่ 3.11 Chat Page Use Case

Use Case:	Chat Page (หน้าสนทนา)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor ต้องเลือกห้องแชทก่อน Actor ต้องมีห้องแชทก่อน
Post-Condition:	Actor สามารถส่งและรับข้อความในห้องแชทได้สำเร็จ
Main Flow:	1. Actor เลือกห้องแชทจาก "Chat Menu" 2. ระบบเปิด Chat Page และแสดงประวัติการสนทนา(ถ้ามี) 3. Actor สามารถดำเนินการดังต่อไปนี้ <ul style="list-style-type: none"> <li>• พิมพ์ข้อความในช่องแชท</li> <li>• กดปุ่ม "ส่ง" เพื่อส่งข้อความ</li> </ul>

	<ul style="list-style-type: none"> <li>แนบรูปภาพหรือเปิดกล้องถ่ายเพื่อส่งในห้องแชท</li> </ul> <p>4. ระบบส่งข้อความและอัปเดตการสนทนาระบบเรียลไทม์</p>
Alternative Flow:	-
Exception Flow:	หาก Actor ต้องการออกจากห้องแชท → กดย้อนกลับหรือปิดหน้าต่างแชท

ตารางที่ 3.12 Send Wound Information in Chat Use Case

Use Case:	Send Wound Information in Chat (ส่งข้อมูลแพลในแชท)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor ต้องอยู่ใน Chat Page Actor ต้องมีข้อมูลแพลที่บันทึกไว้ในระบบ
Post-Condition:	ข้อมูลแพลถูกส่งสำเร็จในห้องแชท
Main Flow:	<ol style="list-style-type: none"> <li>Actor เปิด Chat Page และเลือกห้องแชทที่ต้องการ</li> <li>Actor กดปุ่ม : ด้านขวาบนของหน้า Chat Page</li> <li>ระบบแสดงรายการแพลที่สามารถเลือกส่งได้</li> <li>Actor เลือกแพลที่ต้องการส่ง โดยระบบแสดงข้อมูลแพลดังนี้           <ul style="list-style-type: none"> <li>รูปภาพแพล</li> <li>อวัยวะที่แพลเกิด</li> <li>Label กำกับของแพล</li> </ul> </li> <li>Actor กดที่รูปภาพแพลที่ต้องการส่ง</li> <li>ระบบอปฯ โหลดข้อมูลแพลและรายละเอียดต่างๆ และส่งไปยังห้องแชท</li> </ol>
Alternative Flow:	หาก Actor ไม่พบข้อมูลแพลในระบบ → ระบบจะไม่แสดงรายการแพลเมื่อกด :
Exception Flow:	หาก Actor ต้องการยกเลิกการส่งข้อมูลแพล → สามารถกดปุ่ม “ปิด” หน้าต่างเลือกแพลได้

### ตารางที่ 3.13 Show History of Tracked Wounds Use Case

Use Case:	Show History of Tracked Wounds (แสดงประวัติการติดตามแผล)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor (Nurse) ต้องมีผู้ป่วยในการดูแล Actor ต้องมีข้อมูลแผลที่บันทึกไว้ในระบบ
Post-Condition:	Actor สามารถติดตามแผลที่บันทึกไว้ในระบบได้
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor เข้าสู่ระบบและไปที่หน้า "Follow Up Menu"             <ol style="list-style-type: none"> <li>1.1 Actor (Nurse) เลือกผู้ป่วยที่ต้องการติดตามแผล</li> </ol> </li> <li>2. ระบบแสดงรายการแผลทั้งหมดที่เกิดขึ้นบนร่างกาย (Unique Wound) โดยแสดงข้อมูลของแผลล่าสุด รูปภาพและรายละเอียด</li> <li>3. Actor สามารถเลือกแผลจากรายการที่แสดง เพื่อดูรายละเอียดเพิ่มเติมของแผลนั้น</li> </ol>
Alternative Flow:	หากไม่มีแผลที่ถูกติดตามในระบบ → ระบบจะแจ้งเตือนว่า "ผู้ป่วยไม่มีแผลให้ติดตาม"
Exception Flow:	-

### ตารางที่ 3.14 Show History of the Wound's Diagnosis Use Case

Use Case:	Show History of the Wound's Diagnosis (แสดงประวัติการติดตามแผลนั้นๆ)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว Actor (Nurse) ต้องมีผู้ป่วยในการดูแล Actor ต้องมีข้อมูลแผลที่บันทึกไว้ในระบบ
Post-Condition:	Actor สามารถดูประวัติการติดตามแผลนั้นๆที่บันทึกไว้ในระบบได้
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor เข้าสู่ระบบและไปที่หน้า "Follow Up Menu"</li> </ol>

	<p>1.1 Actor (Nurse) เลือกผู้ป่วยที่ต้องการติดตามแพลด</p> <p>2. Actor เลือกแพลด (Unique Wound) จากรายการแพลดที่ได้รับการติดตาม</p> <p>3. ระบบแสดงประวัติการตรวจของแพลดที่เลือก โดยแสดงข้อมูลการตรวจทั้งหมดของแพลดนี้ เช่น หากแพลดที่เลือกมีการตรวจแล้ว 3 ครั้ง ระบบจะแสดงรายการการตรวจทั้งหมดให้เลือกเข้าไปดูรายละเอียดของแต่ละครั้ง</p> <p>4. Actor เลือกแพลดที่ต้องการดูรายละเอียดเพิ่มเติม</p>
Alternative Flow:	-
Exception Flow:	หาก Actor ไม่ต้องการดูรายละเอียดของแพลด → สามารถคลิกสัญลักษณ์ “ลูกศรย้อนกลับ” เพื่อย้อนกลับไปหน้าก่อนได้

ตารางที่ 3.15 Show Profile Use Case

Use Case:	Show Profile (แสดงข้อมูลโปรไฟล์)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว
Post-Condition:	Actor สามารถดูข้อมูลโปรไฟล์ของตนเองได้
Main Flow:	<p>1. Actor เข้าสู่ระบบและไปที่หน้า "Profile"</p> <p>2. ระบบแสดงข้อมูลโปรไฟล์ของ Actor รวมถึงข้อมูลส่วนตัว</p> <ul style="list-style-type: none"> <li>• ชื่อ-นามสกุล</li> <li>• เพศ</li> <li>• วัน/เดือน/ปีเกิด</li> <li>• เลขบัตรประชาชน</li> <li>• หมายเลขโทรศัพท์</li> </ul> <p>3. Actor สามารถดูข้อมูลโปรไฟล์ทั้งหมดที่ระบบแสดงให้</p>
Alternative Flow:	-
Exception Flow:	-

ตารางที่ 3.16 Edit Profile Use Case

Use Case:	Edit Profile (แก้ไขข้อมูลโปรไฟล์)
Actor:	Patient, Nurse
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว
Post-Condition:	Actor สามารถแก้ไขข้อมูลโปรไฟล์และบันทึกข้อมูลใหม่ได้
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor เข้าสู่ระบบและไปที่หน้า "Profile"</li> <li>2. ระบบแสดงข้อมูลโปรไฟล์ของ Actor รวมถึงข้อมูลส่วนตัว <ul style="list-style-type: none"> <li>• ชื่อ-นามสกุล</li> <li>• เพศ</li> <li>• วัน/เดือน/ปีเกิด</li> <li>• เลขบัตรประชาชน</li> <li>• หมายเลขโทรศัพท์</li> </ul> </li> <li>3. Actor สามารถดูข้อมูลโปรไฟล์ทั้งหมดที่ระบบแสดงให้</li> </ol>
Alternative Flow:	หาก Actor ไม่ต้องการแก้ไขข้อมูลและกด "ลูกศรย้อนกลับ" → ระบบจะไม่ทำการบันทึกการเปลี่ยนแปลงใดๆ และแสดงข้อมูลเดิมที่ไม่เปลี่ยนแปลง
Exception Flow:	หากมีการกรอกข้อมูลที่ไม่ถูกต้อง → ระบบจะแจ้งเตือนว่า "กรุณากรอกข้อมูลให้ถูกต้อง"

ตารางที่ 3.17 View Dashboard Use Case

Use Case:	View Dashboard (ดูแดชบอร์ด)
Actor:	Nurse, Administrator
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว
Post-Condition:	Actor สามารถดูข้อมูลภาพรวมของระบบได้
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor ไปที่แถบ Sidebar ชื่อ “แดชบอร์ด”</li> <li>2. ระบบแสดงข้อมูลแดชบอร์ดให้ Actor</li> </ol>

	<ul style="list-style-type: none"> <li>แสดงจำนวนผู้ป่วยในระบบ</li> <li>แสดงจำนวนผู้ป่วยที่ได้รับการรักษาแล้ว</li> <li>แสดงจำนวนผู้ป่วยที่ยังไม่ได้รับการรักษา</li> <li>แสดงจำนวนของแพลตฟอร์มต่อละดับ</li> <li>แสดงจำนวนแพลตฟอร์มตามอาการ</li> <li>ผู้ป่วยในระบบล่าสุด</li> </ul>
AlternativeFlow:	-
Exception Flow:	-

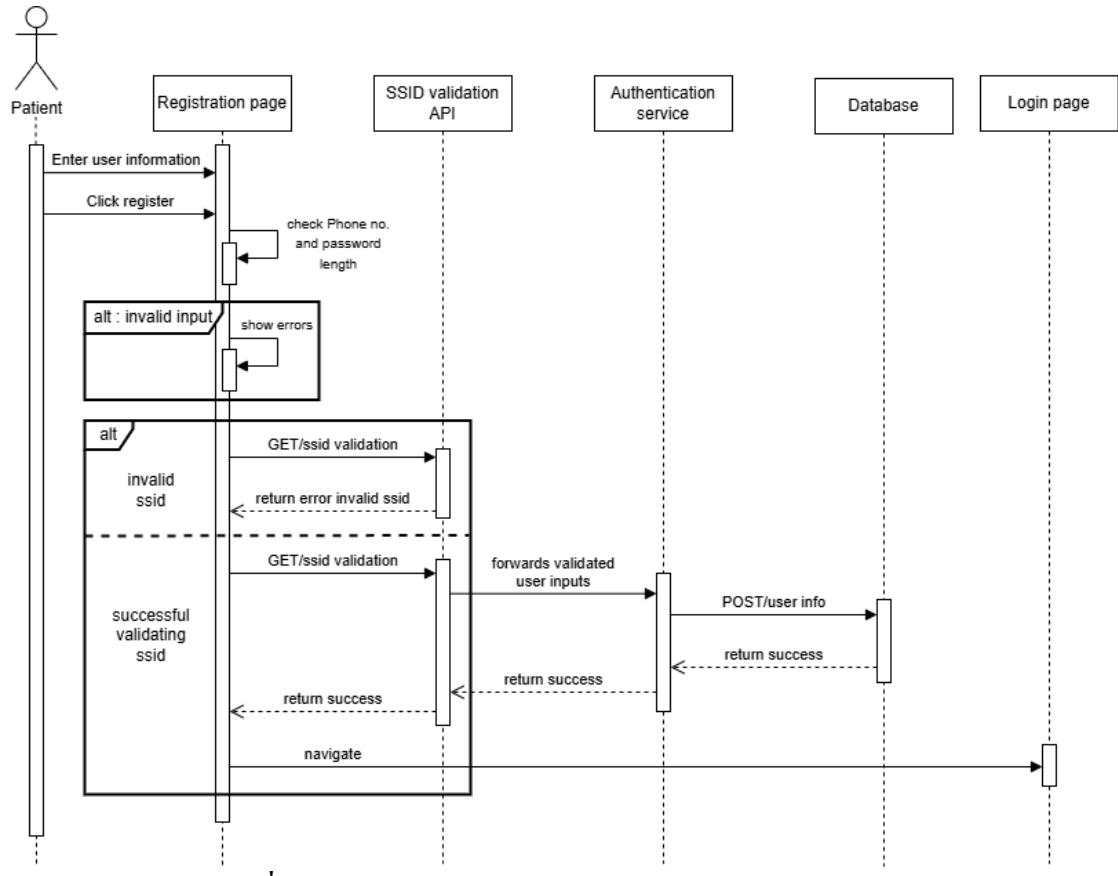
ตารางที่ 3.18 Manage Patient Use Case

Use Case:	Manage Patient (ขัดการผู้ป่วย)
Actor:	Nurse, Administrator
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว
Post-Condition:	Actor สามารถเพิ่ม ลบ หรือแก้ไขข้อมูลผู้ป่วยได้
Main Flow:	<ol style="list-style-type: none"> <li>Actor ไปที่แดบ Sidebar ชื่อ “ผู้ป่วย”</li> <li>Actor สามารถเลือกเพิ่ม แก้ไข หรือลบผู้ป่วยได้</li> <li>ระบบบันทึกการเปลี่ยนแปลงลงฐานข้อมูล</li> </ol>
AlternativeFlow:	ถ้า Actor ยกเลิกการแก้ไข ระบบจะไม่เปลี่ยนแปลงข้อมูล
Exception Flow:	-

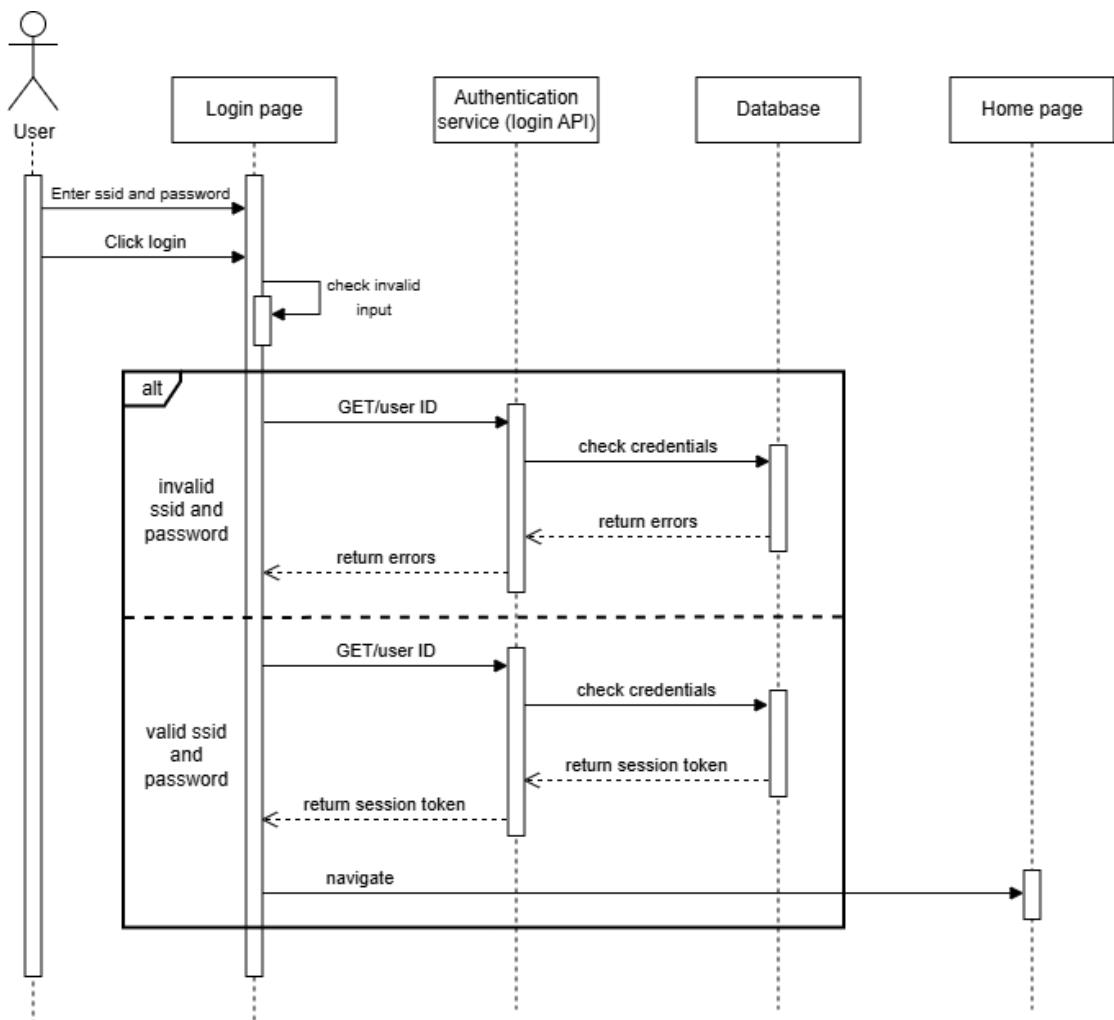
### ตารางที่ 3.19 Manage Nurse Use Case

Use Case:	Manage Nurse (จัดการพยาบาล)
Actor:	Administrator
Pre-Condition:	Actor ต้องเข้าสู่ระบบแล้ว
Post-Condition:	Actor สามารถเพิ่ม ลบ หรือแก้ไขข้อมูลพยาบาลได้
Main Flow:	<ol style="list-style-type: none"> <li>1. Actor ไปที่แท็บ Sidebar ชื่อ “พยาบาล”</li> <li>2. Actor สามารถเลือกเพิ่ม แก้ไข หรือลบพยาบาลได้</li> <li>3. ระบบบันทึกการเปลี่ยนแปลงลงฐานข้อมูล</li> </ol>
AlternativeFlow:	ถ้า Actor ยกเลิกการแก้ไข ระบบจะไม่เปลี่ยนแปลงข้อมูล
Exception Flow:	-

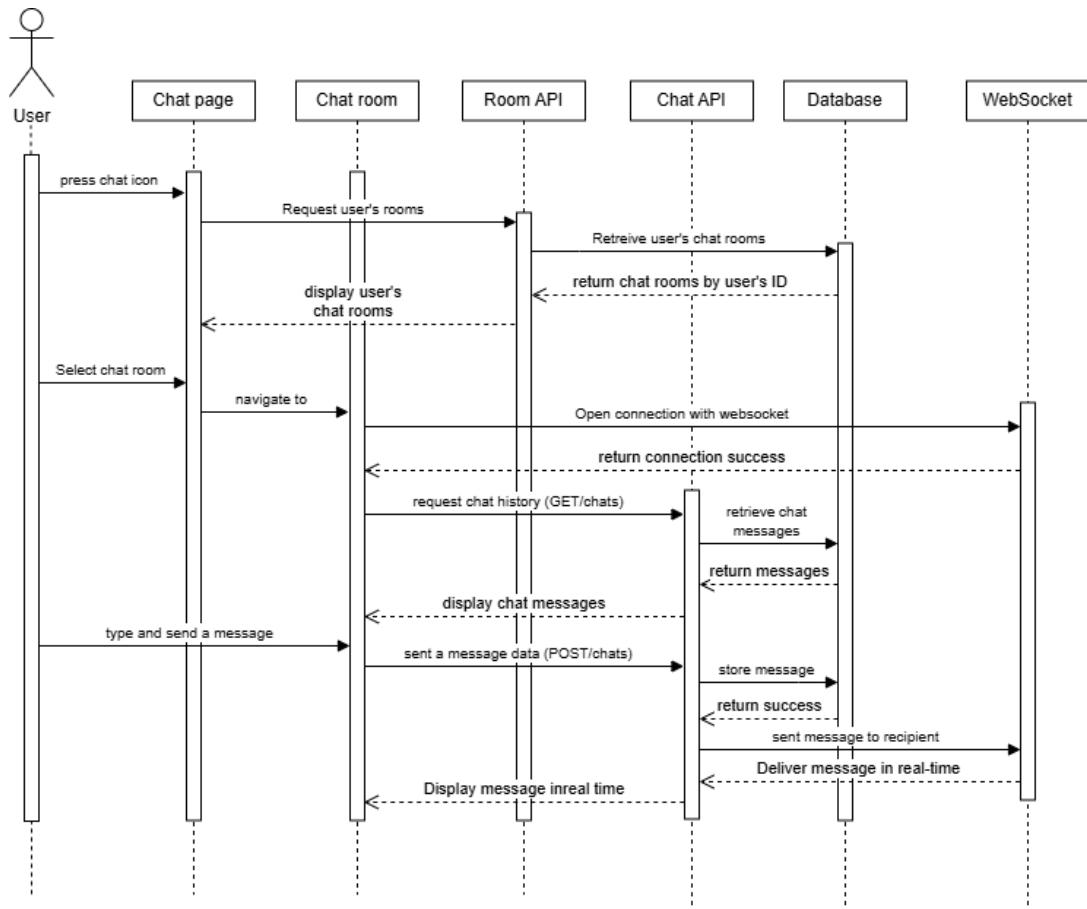
### 3.4 การออกแบบ Sequence Diagram



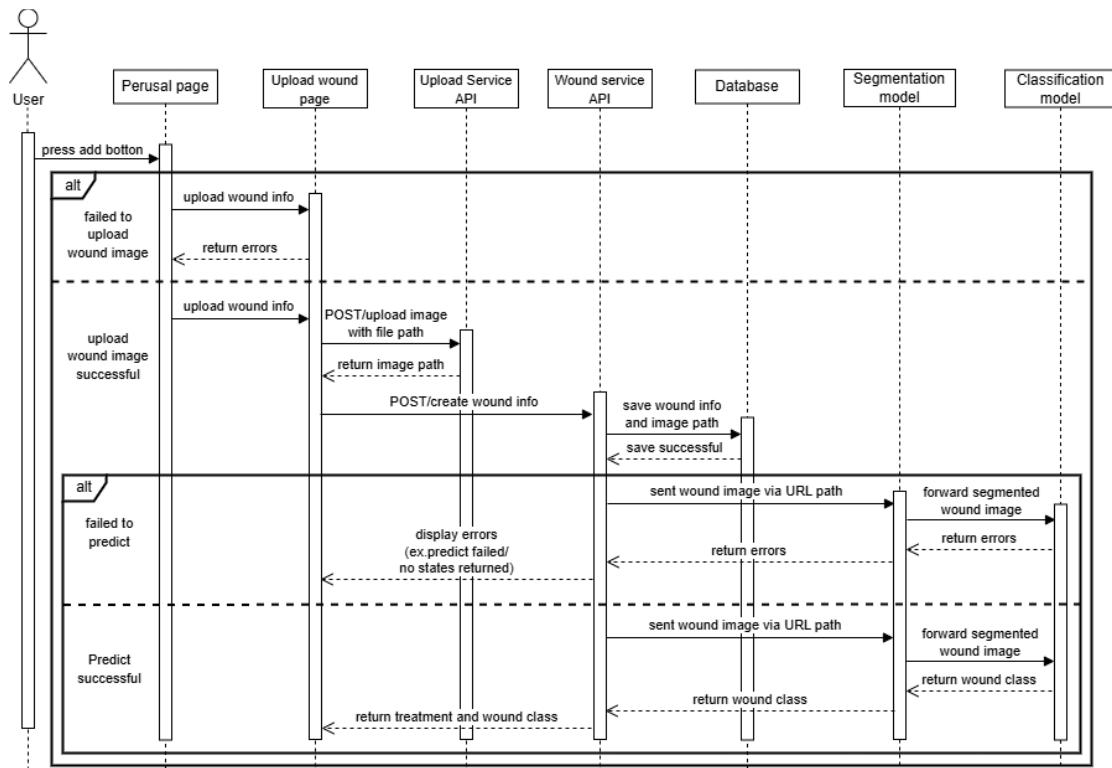
รูปที่ 3.6 ลำดับการทำงานในการลงทะเบียนผู้ใช้สำหรับผู้ป่วย



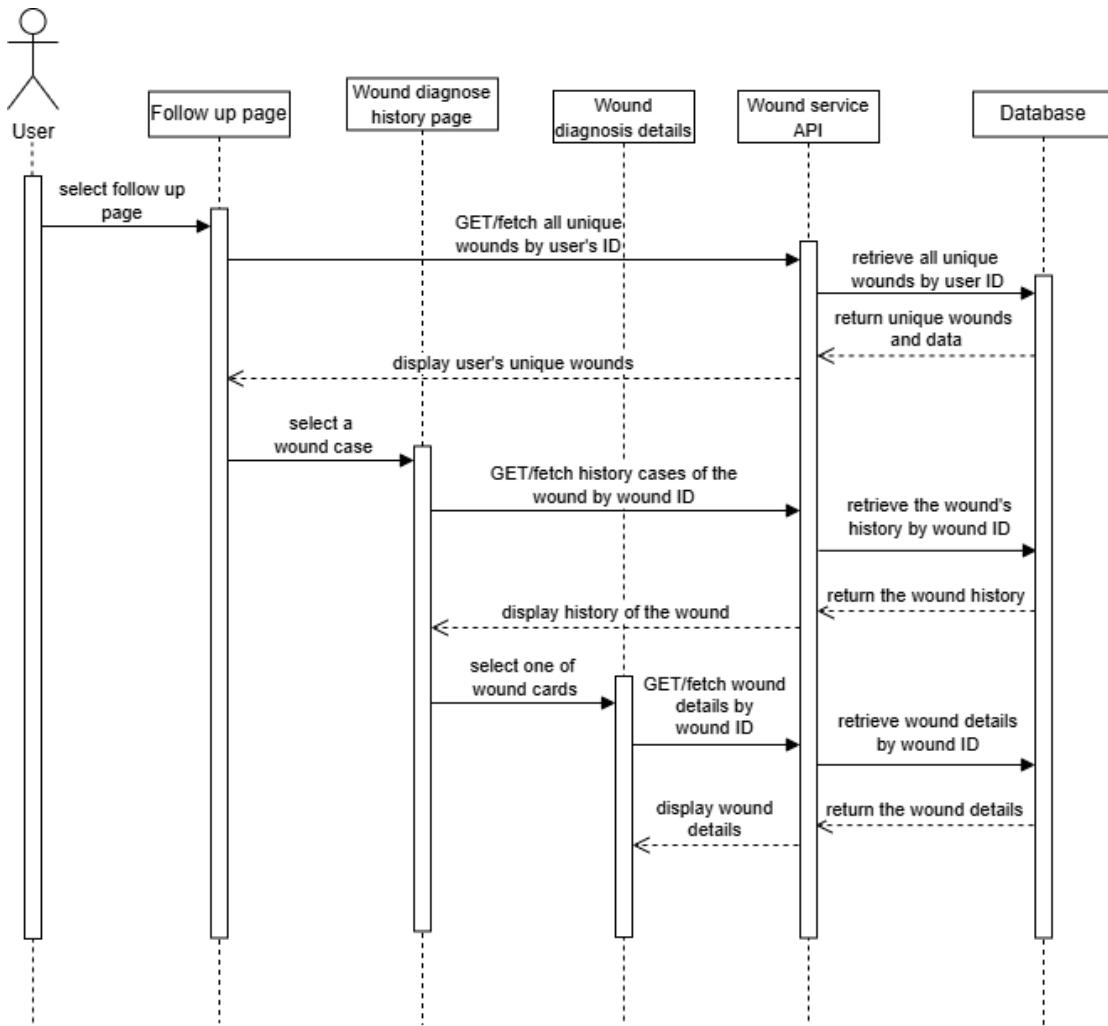
รูปที่ 3.7 ลำดับการทำงานในการเข้าสู่ระบบ



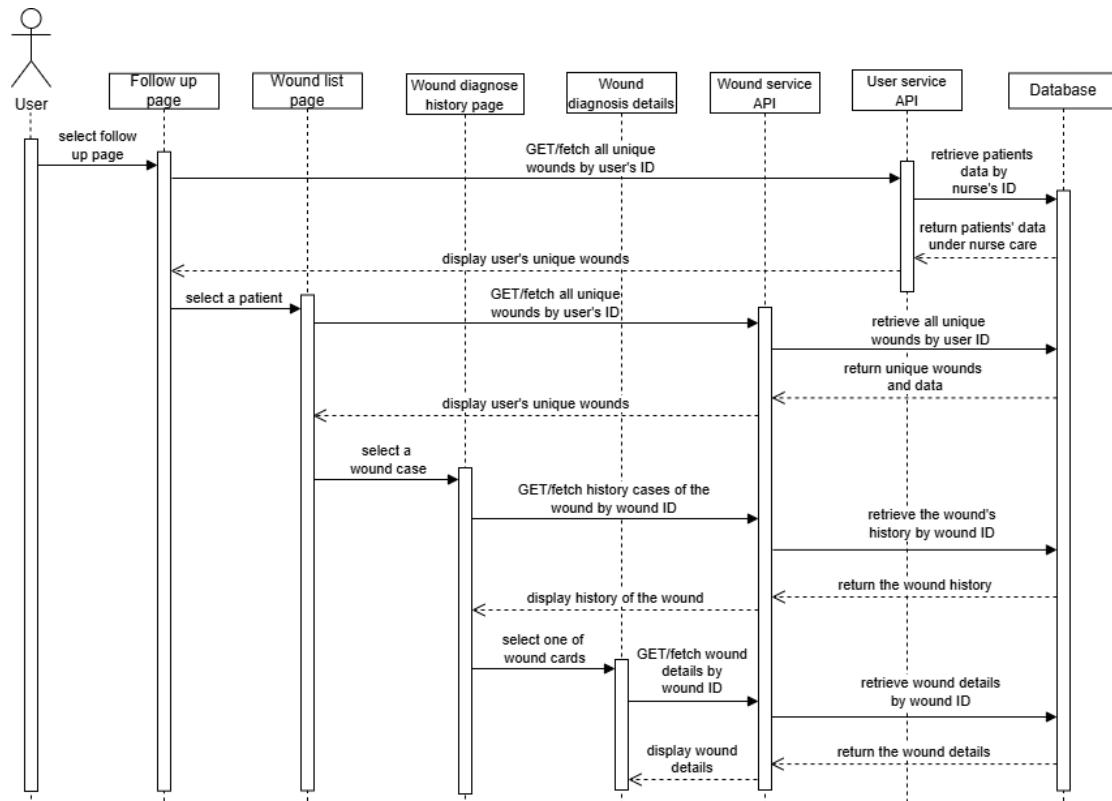
รูปที่ 3.8 ลำดับการทำงานในระบบการแชทของผู้ใช้งานระบบ



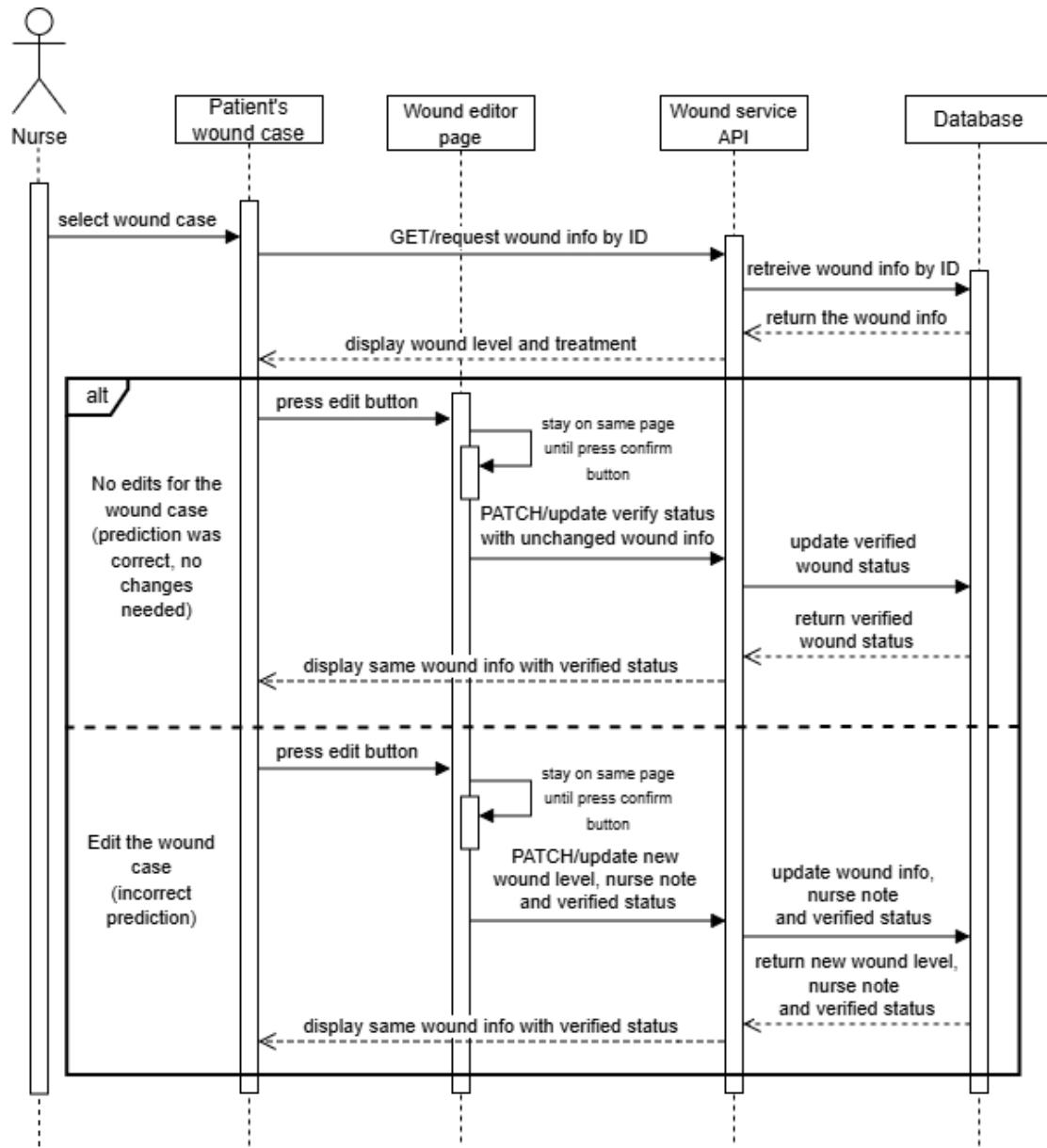
รูปที่ 3.9 ลำดับการทำงานในระบบการเพิ่มแพลเจ้าสุร์ระบบเพื่อแสดงผลการทำนาย



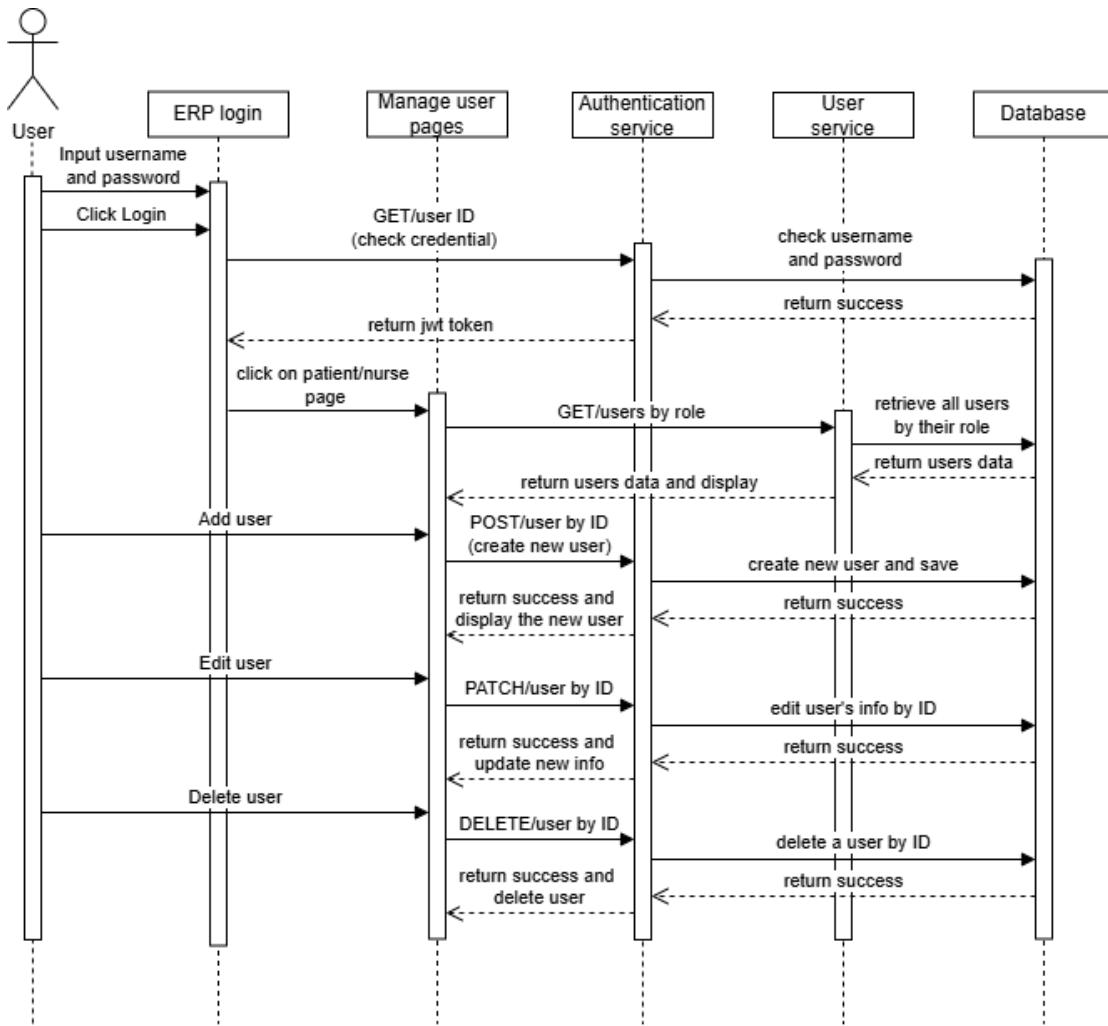
รูปที่ 3.10 ลำดับการทำงานในระบบติดตามอาการของผู้ป่วย



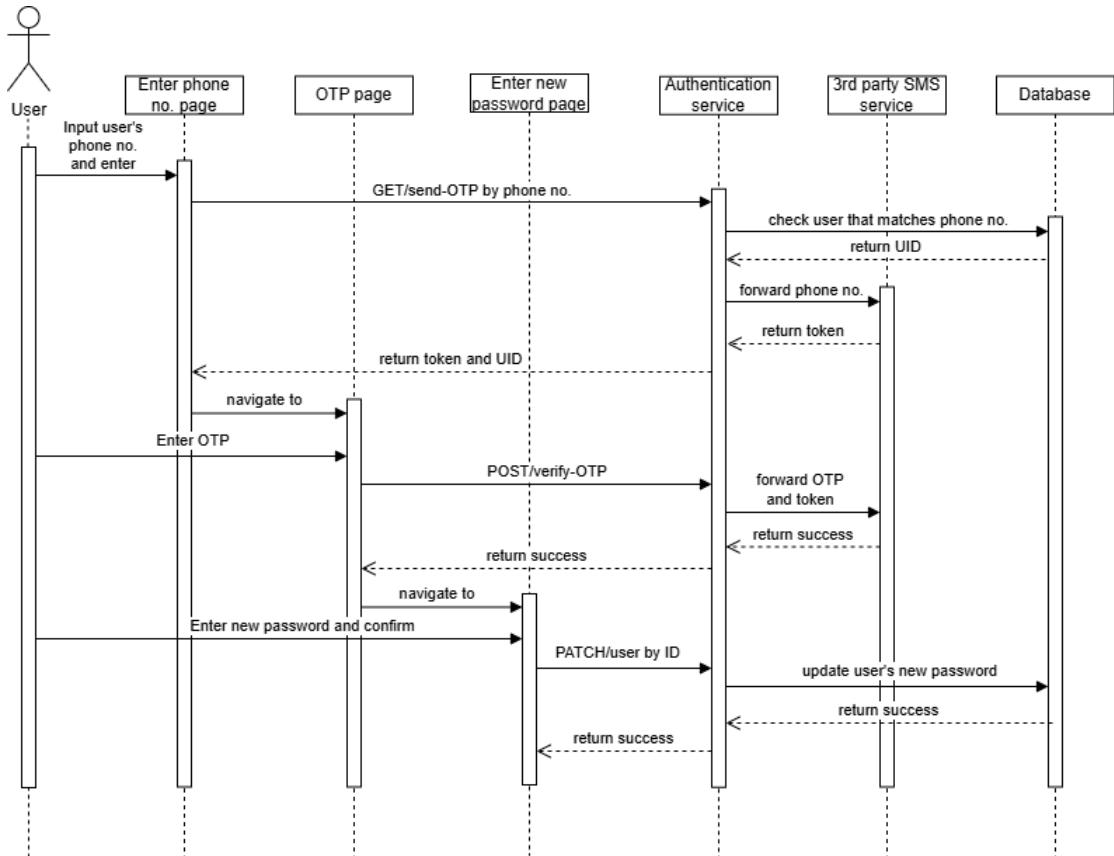
รูปที่ 3.11 ลำดับการทำงานในระบบติดตามอาการของผู้ป่วยจากพยาบาล



รูปที่ 3.12 ลำดับการทำงานในระบบการตรวจสอบความถูกต้องผลโดยพยาบาล



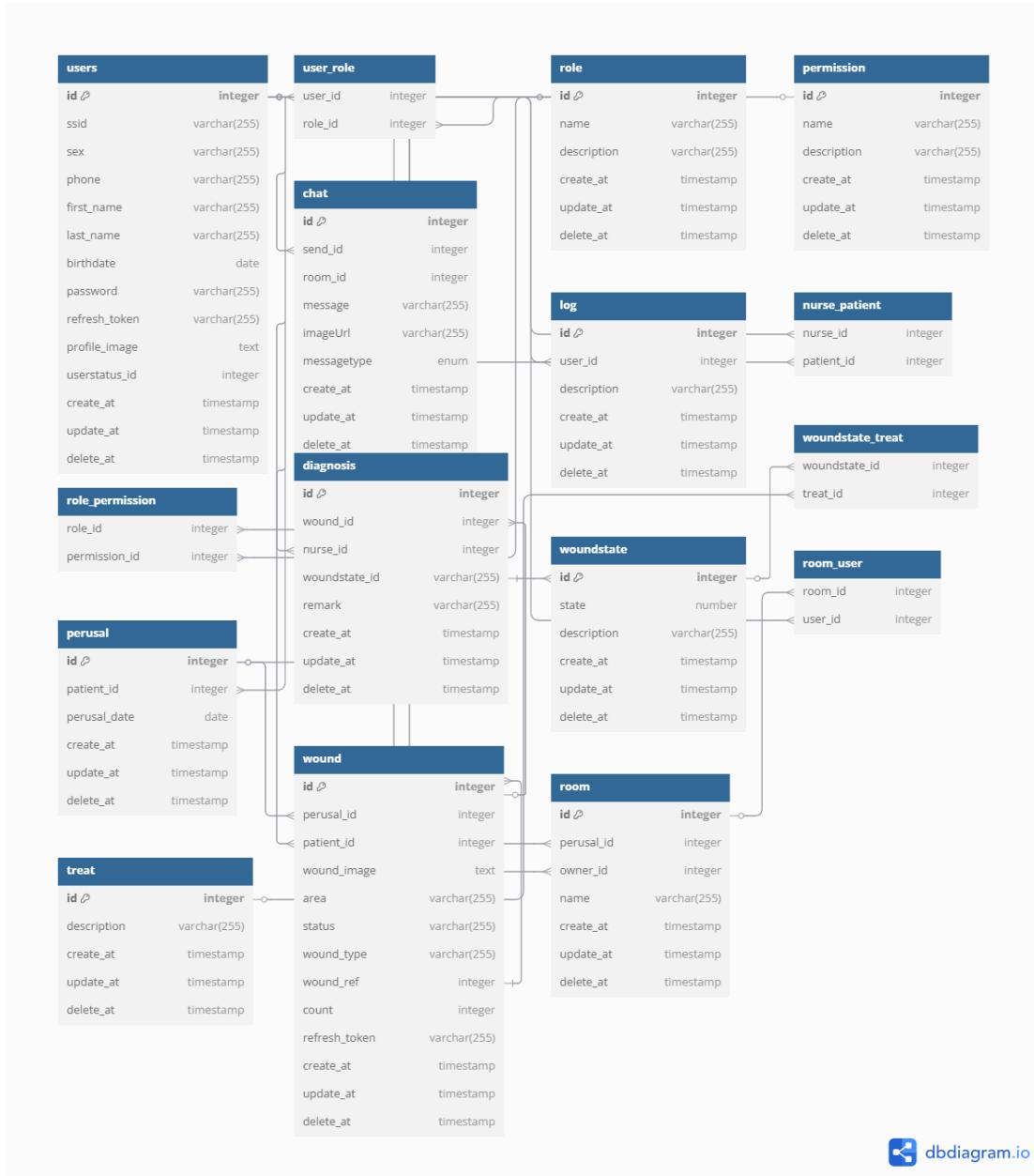
รูปที่ 3.13 ลำดับการทำงานในระบบ ERP



รูปที่ 3.14 ลำดับการทำงานในระบบส่ง OTP เพื่อกรณีลืมรหัสผ่าน

## 3.5 การออกแบบฐานข้อมูล

### 3.5.1 การออกแบบ Entity Relationship Diagram



รูปที่ 3.15 ภาพ ERD Diagram

ตารางที่ 3.9 รายละเอียดตารางผู้ใช้ (User)

Attribute	Type	Note	Description
id	integer	pk	รหัสผู้ใช้ (Primary Key)
ssid	varchar(255)	unique	หมายเลขประจำตัวประชาชนของผู้ใช้ ซึ่งต้องไม่ซ้ำกัน
sex	varchar(255)		เพศของผู้ใช้
phone	varchar(255)	unique	เบอร์โทรศัพท์ของผู้ใช้
first_name	varchar(255)		ชื่อจริงของผู้ใช้
last_name	varchar(255)		นามสกุลของผู้ใช้
birthdate	date		วันเกิดของผู้ใช้
password	varchar(255)	hashing โดย argon2	รหัสผ่านที่เข้ารหัสเพื่อการยืนยันตัวตน ที่ได้รับการ hashing
refresh_token	varchar(255)	hashing โดย argon2	refresh_token ใช้ต่ออายุ access_token ที่ได้รับการ hashing
create_at	timestamp		เวลาที่สร้างข้อมูลผู้ใช้
update_at	timestamp		เวลาที่อัปเดตข้อมูลผู้ใช้ล่าสุด
delete_at	timestamp		เวลาที่ลบข้อมูลผู้ใช้อย่างไม่ถาวร (soft delete)

ตารางที่ 3.10 รายละเอียดตารางสิทธิ์ (Role)

Attribute	Type	Note	Description
id	integer	pk	รหัสบันทึก (Primary Key)
name	varchar(255)		ชื่อของบทบาท เช่น คนไข้ พยาบาล ผู้ดูแลระบบ
description	varchar(255)		คำอธิบายเกี่ยวกับบทบาทนี้
create_at	timestamp		เวลาที่สร้างบทบาทนี้
update_at	timestamp		เวลาที่อัปเดตข้อมูลบทบาทล่าสุด
delete_at	timestamp		เวลาที่ลบบทบาทอย่างไม่ถาวร (soft delete)

ตารางที่ 3.11 รายละเอียดตารางสิทธิ์ (Permission)

Attribute	Type	Note	Description
id	integer	pk	รหัสสิทธิ์ (Primary Key)
name	varchar(255)		ชื่อของสิทธิ์ เช่น การอ่าน การเขียน การแก้ไข
description	varchar(255)		คำอธิบายเกี่ยวกับสิทธินี้
create_at	timestamp		เวลาที่สร้างสิทธินี้
update_at	timestamp		เวลาที่อัปเดตข้อมูลสิทธิ์ล่าสุด
delete_at	timestamp		เวลาที่ลบสิทธิ์อย่างไม่ถาวร (soft delete)

ตารางที่ 3.12 รายละเอียดตารางระบบแชท (Chat)

Attribute	Type	Note	Description
id	integer	pk	รหัสข้อความ (Primary Key)
send_id	integer	fk	รหัสผู้ส่งข้อความ (เชื่อมโยงไปยังผู้ใช้ในตาราง user)
receive_id	integer	fk	รหัสผู้รับข้อความ (เชื่อมโยงไปยังผู้ใช้ในตาราง user)
message type	enum messagetype	Text หรือ Image	ชนิดของข้อมูลที่ส่งใน chat
message	varchar(255)	เป็น null กรณีที่ message type เป็น Image	ข้อความที่ถูกส่ง
ImageUrl	varchar(255)	เป็น null กรณีที่ message type เป็น Text	Image path ของรูปที่ส่งใน chat
create_at	timestamp		เวลาที่ส่งข้อความ
update_at	timestamp		เวลาที่อัปเดตข้อความล่าสุด
delete_at	timestamp		เวลาที่ลบข้อความอย่างไม่ถาวร (soft delete)

ตารางที่ 3.13 รายละเอียดตารางระบบล็อก (Log)

Attribute	Type	Note	Description
id	integer	pk	รหัสล็อก (Primary Key)
user_id	integer	fk	รหัสผู้ใช้ (เชื่อมโยงไปยังตาราง user)
description	varchar(255)		รายละเอียดของเหตุการณ์หรือกิจกรรมที่บันทึกลงในล็อก
create_at	timestamp		เวลาที่สร้างบันทึกล็อก
update_at	timestamp		เวลาที่อัปเดตบันทึกล่าสุด
delete_at	timestamp		เวลาที่ลบบันทึกอย่างไม่ถาวร (soft delete)

ตารางที่ 3.14 รายละเอียดตารางการตรวจ (perusal)

Attribute	Type	Note	Description
id	integer	pk	รหัสการตรวจ (Primary Key)
patient_id	integer	fk	รหัสผู้ป่วย (เชื่อมโยงไปยังตาราง user)
perusal_date	date		วันที่ทำการตรวจสอบหรือพิจารณาผู้ป่วย
create_at	timestamp		เวลาที่สร้างการบันทึกนี้
update_at	timestamp		เวลาที่อัปเดตข้อมูลการตรวจสอบล่าสุด
delete_at	timestamp		เวลาที่ลบการบันทึกอย่างไม่ถาวร (soft delete)

ตารางที่ 3.15 รายละเอียดตารางการวินิจฉัย (diagnosis)

Attribute	Type	Note	Description
id	integer	pk	รหัสการวินิจฉัย (Primary Key)
wound_id	integer	fk	รหัสแพล (เชื่อมโยงไปยังตาราง wound)
perusal_date	date		รหัสการตรวจ (เชื่อมโยงไปยัง ตาราง perusal)
treat	varchar(255)		รายละเอียดหรือแผนการรักษาที่ กำหนด
create_at	timestamp		เวลาที่สร้างการบันทึกการวินิจฉัย
update_at	timestamp		เวลาที่อัปเดตข้อมูลการวินิจฉัย ล่าสุด
delete_at	timestamp		เวลาที่ลบการบันทึกการวินิจฉัย อย่างไม่ถาวร (soft delete)

ตารางที่ 3.16 รายละเอียดตารางแพล (wound)

Attribute	Type	Note	Description
id	integer	pk	รหัสแพล (Primary Key)
perusal_id	integer	fk	รหัสการตรวจ (เชื่อมโยงไปยัง ตาราง perusal)
wound_image	text		Path ที่เก็บรูปแพล
area	varchar(255)		พื้นที่ของแพล เช่น แขน, ขา, หรือ ส่วนอื่น ๆ
status	enum status	ตรวจแล้ว หรือ รอตรวจ	สถานะของแพล

wound_type	enum wound_ty pe	แผลใหม่ หรือ แผลเก่า	ชนิดของแผลนั้น
count	integer		ครั้งที่ของแผลนั้นในของแต่ละผู้ใช้
create_at	timestamp		เวลาที่สร้างข้อมูลแผลนี้
update_at	timestamp		เวลาที่อัปเดตข้อมูลแผลล่าสุด
delete_at	timestamp		เวลาที่ลบข้อมูลแผลอย่างไม่ถาวร (soft delete)
perusal_id	integer	fk	รหัสการตรวจ (เชื่อมโยงไปยังตาราง perusal)

ตารางที่ 3.17 รายละเอียดตารางระดับแผล (woundstate)

Attribute	Type	Note	Description
id	integer	pk	รหัสระดับแผล (Primary Key)
state	integer		ระดับของแผล
description	varchar(2 55)		รายละเอียดของระดับแผล
create_at	timestamp		เวลาที่สร้างข้อมูลระดับแผลนี้
update_at	timestamp		เวลาที่อัปเดตข้อมูลระดับแผลล่าสุด
delete_at	timestamp		เวลาที่ลบข้อมูลระดับแผลอย่างไม่ถาวร (soft delete)

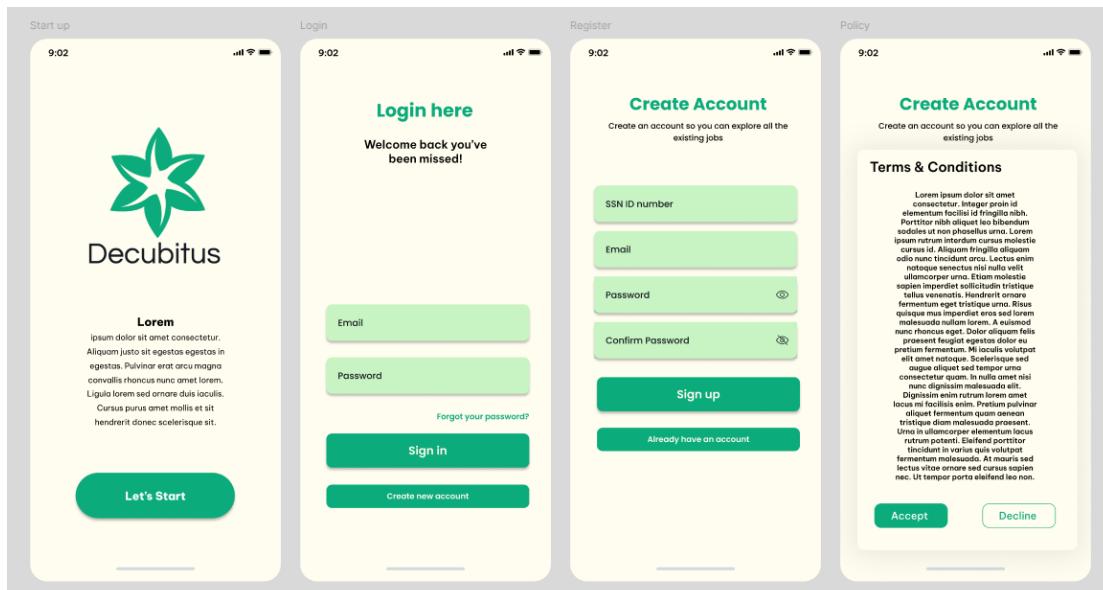
ตารางที่ 3.18 รายละเอียดตารางวิธีการรักษา (treat)

Attribute	Type	Note	Description
id	integer	pk	รหัสการรักษา (Primary Key)
description	varchar(255)		รายละเอียดของรักษา
create_at	timestamp		เวลาที่สร้างข้อมูลการรักษา
update_at	timestamp		เวลาที่อัปเดตข้อมูลการรักษาล่าสุด
delete_at	timestamp		เวลาที่ลบข้อมูลการรักษาอย่างไม่ถาวร (soft delete)

## 3.6 การออกแบบ UI

### 3.6.1 ระบบการยืนยันตัวตน (Authentication)

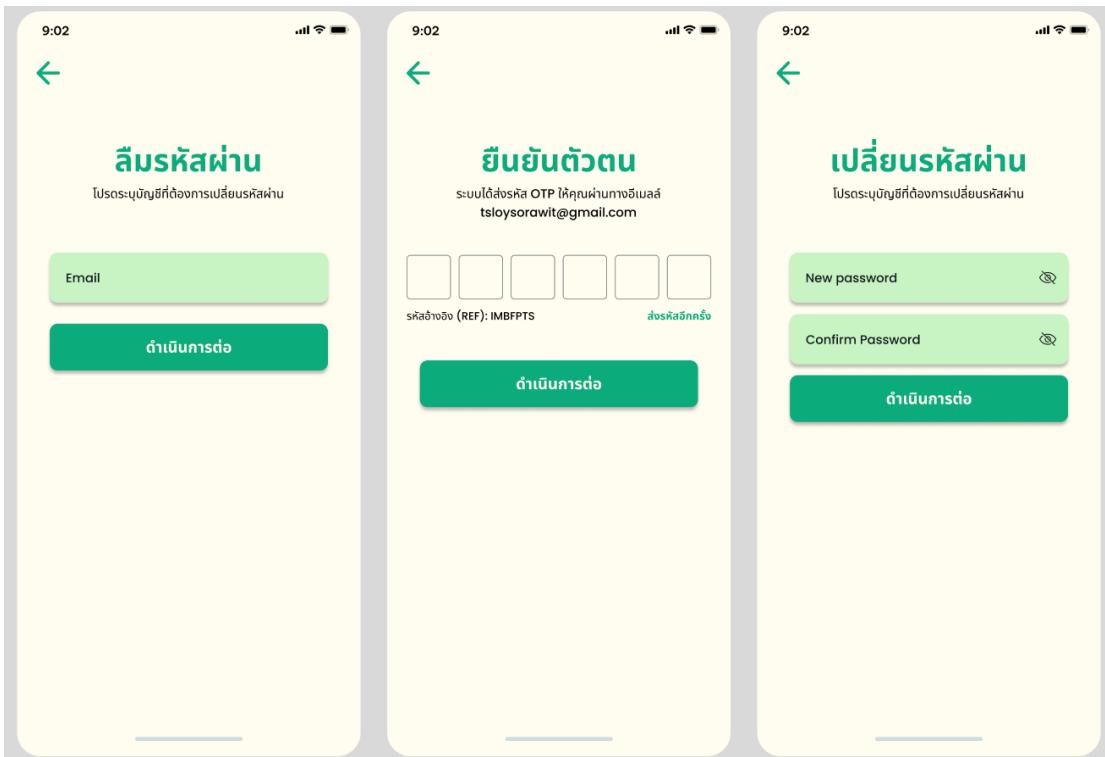
จัดทำหน้าการสมัครสมาชิกและการเข้าสู่ระบบ โดยผู้เริ่มต้นใช้งานครั้งแรกต้องลงทะเบียนสร้างบัญชี กดยอมรับเงื่อนไขการใช้งานและคำยินยอม



รูปที่ 3.16 หน้าแนะนำแอปพลิเคชัน หน้าเข้าสู่ระบบ สมัครสมาชิก และหน้าเงื่อนไขการใช้งานแอป พลิกะชัน

### 3.6.2 ระบบหน้าระบบลืมรหัสและการรีเซ็ตรหัสผ่าน

จัดทำหน้าลืมรหัสผ่านสำหรับผู้ใช้ทั่วไปที่ลืมรหัสเพื่อเข้าระบบตนเอง รหัส OTP จะถูกส่งไปที่อีเมลที่ถูกกรอก เมื่อกรอกรหัส OTP ถูกต้องก็จะไปยังหน้าเปลี่ยนรหัสผ่าน



รูปที่ 3.17 หน้าระบบลืมรหัสและการรีเซ็ตรหัสผ่าน

### 3.6.3 ระบบการทำรายการตรวจสอบแพลตฟอร์ม

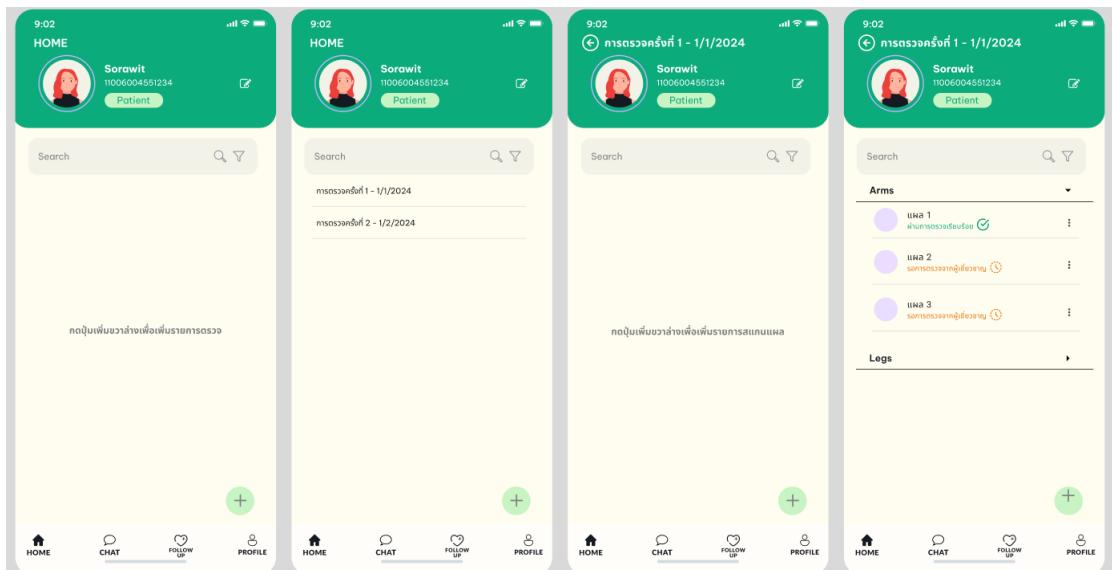
จัดทำหน้ารายการตรวจสอบซึ่งเป็นหน้าหลักของผู้ใช้งานทั่วไป โดยที่เมื่อที่ผู้ใช้งานต้องการสแกนแพลตฟอร์มใดๆ ขั้นแรกต้องเพิ่มรายการตรวจสอบก่อนด้วยการกดปุ่มเพิ่มรายการทางด้านขวาล่างของจอ



ຮູບທີ 3.18 ມໍາຮັບຮັບການທໍາຮາຍການຕຽບແລ້ວ

#### 3.6.4 ຮະບນສ້າງຮາຍການແພດຂອງຜູ້ໃຊ້ຈານທ້າໄປ

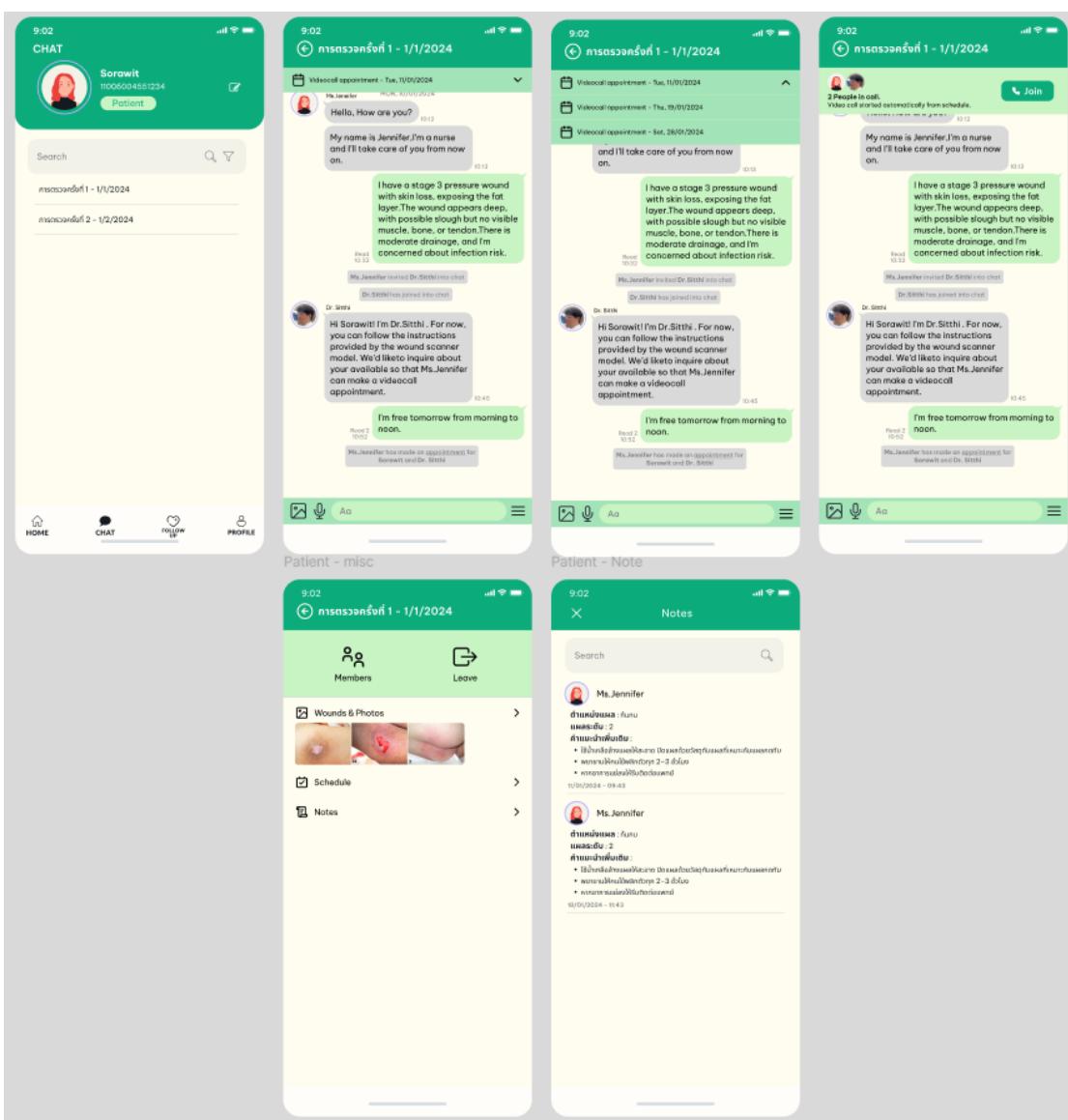
ຈັດທຳໜ້າການສ້າງຮາຍການແພດຂອງຜູ້ໃຊ້ຈານທ້າໄປ ເມື່ອເຂົ້າມາໃນຮາຍການຕຽບແລ້ວ ກົດປຸ່ມ  
ຂວາລ່າງ ຈະເປັນການສ້າງຮາຍການແພດ ໂດຍເບື້ອງຕັ້ນຜູ້ໃຊ້ຕ້ອງກຣອກຂໍ້ມູນທີ່ຈຳເປັນເຊັ່ນ ຮູບກາພແພດ  
ຕໍ່ແໜ່ງແພດ ເປັນຕົ້ນ ເມື່ອກຣອກຂໍ້ມູນເສົ່າງສິ້ນ ໄກສົ່ງປຸ່ມດຳເນີນການຕ່ອງເພື່ອສ່າງຂໍ້ມູນໄໝໄມເດລ  
ປະເມີນແພດ ຈຶ່ງພຸລັພີ່ທີ່ຈະໄດ້ກໍ່ຮາຍການແພດທີ່ມີບັນດັບຕອນກາຮູ້ແພດເມື່ອນັ້ນ ທີ່ນີ້ຮາຍການແພດ  
ດັ່ງກ່າວຈະຄຸກເກີນໄວ້ໃນຮາຍການຕຽບແລ້ວ ດັ່ງກ່າວຈະຄຸກສ່າງຕ່ອງໄກ້ກັບພາຍາລເພື່ອຕຽບສອບຄວາມຄຸກຕ້ອງ<sup>2</sup>  
ແລ້ວ ໄກສົ່ງປຸ່ມນຳເພີ້ມເດີມ



ຮູບທີ 3.19 ມໍາຮັບຮັບການແພດຂອງໃນການຕຽບແຕ່ລະຄົ້ງ

### 3.6.5 ระบบแชทของผู้ใช้งานทั่วไป

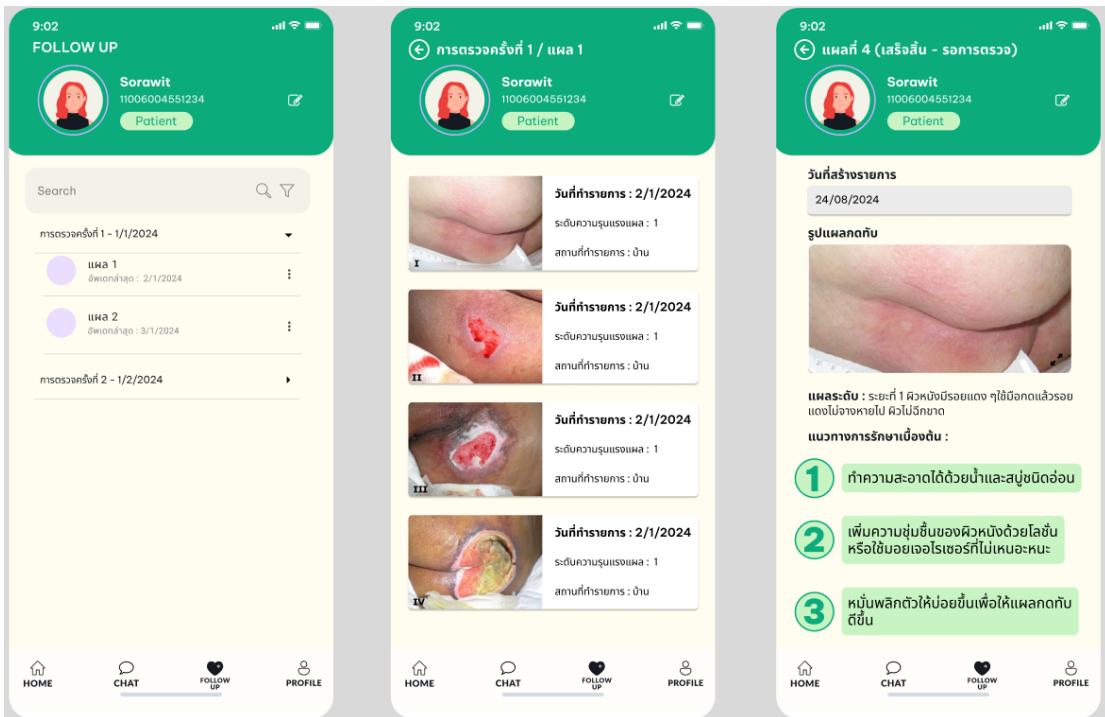
จัดทำหน้าแดชบอร์ดของผู้ใช้งานทั่วไป การที่ผู้ใช้งานจะใช้งานระบบแชทนี้ได้จะต้องมีพยาบาลรับผู้ใช้งานเข้าในการดูแลก่อน โดยสำหรับผู้ใช้งานครั้งแรกสามารถสังเกตจากสถานะการยืนยันแพลตฟอร์มได้ว่ามีการยืนยันแล้วหรือไม่ และจะต้องมีอย่างน้อย 1 แพลตฟอร์มในการตรวจครั้งเดียวที่ได้รับการยืนยันแล้ว จำนวนห้องแชทจะมีตามจำนวนรายการตรวจ โดยเป็นต้นระบบแชทมีฟังก์ชันการสนทนาทั่วไป สามารถส่งภาพและ video call ได้ สามารถดูรูปแพลตฟอร์มของรายการตรวจครั้งนั้นได้ มีระบบ note ที่พยาบาลและหมอสามารถเพิ่มเติมให้ผู้ใช้ได้



รูปที่ 3.20 หน้าระบบแชทของผู้ใช้งานทั้งผู้ป่วย หมอ และพยาบาล

### 3.6.6 ระบบการติดตามแพลงของผู้ใช้งานทั่วไป

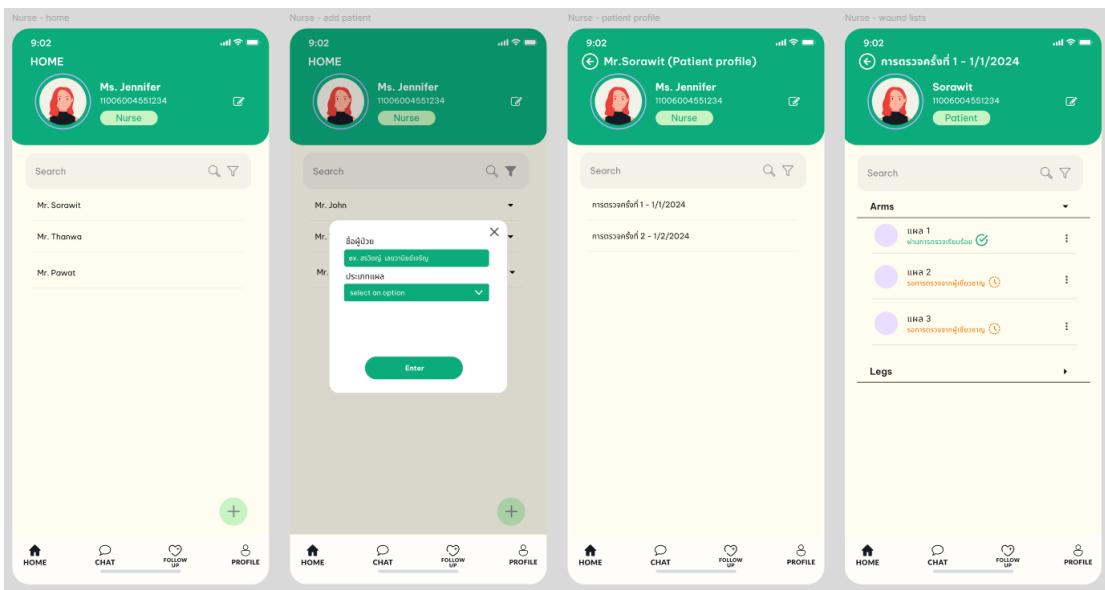
จัดทำหน้าระบบการติดตามแพลงของผู้ใช้งานทั่วไป โดยหน้าดังกล่าวจะแสดงประวัติของแพลงที่รักษาหายแล้วในแต่ละครั้งซึ่งสามารถกดเข้าไปดูประวัติการประมวลผลของโน๊มเดลในแต่ละรายการแพลงได้



รูปที่ 3.21 หน้าระบบการติดตามแพลงของผู้ใช้งานทั่วไป (ผู้ป่วย)

### 3.6.7 ระบบการเพิ่มผู้ป่วยในการดูแลของพยาบาล

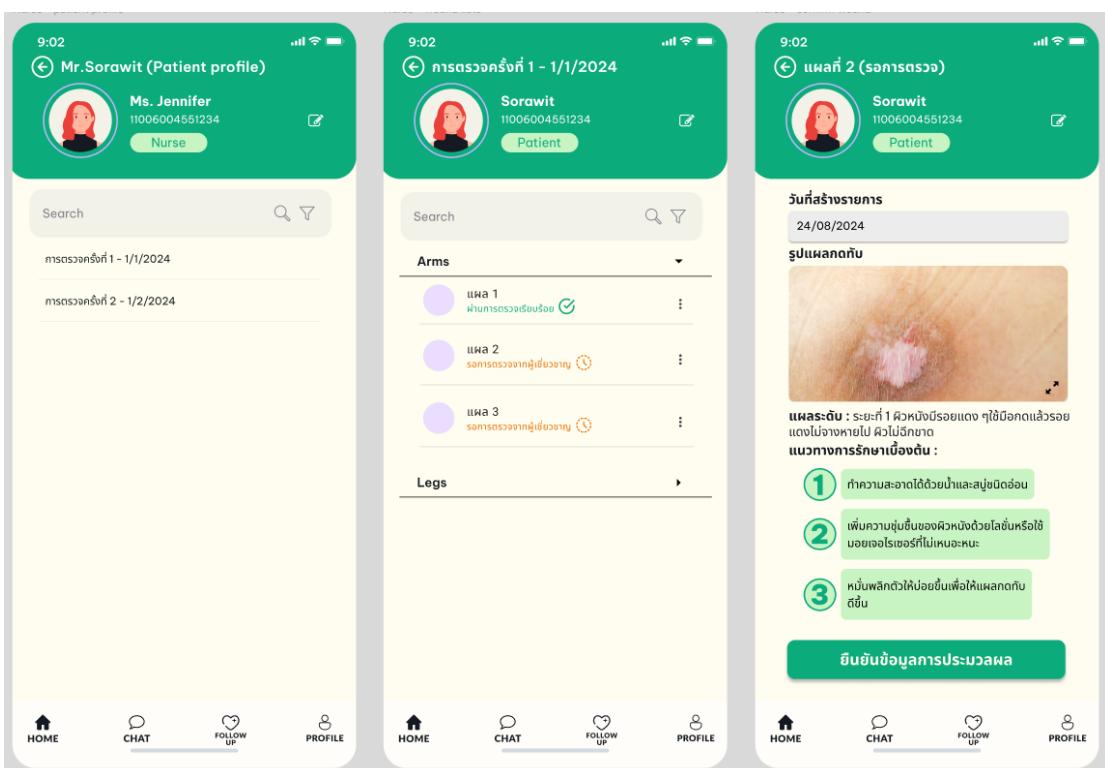
จัดทำหน้าส่วนการเพิ่มผู้ป่วยในการดูแลของพยาบาล ซึ่งเป็นหน้าหลักของผู้ใช้ที่เป็นพยาบาล สามารถเพิ่มได้โดยกดปุ่มขวาล่างเพื่อหาผู้ป่วย เมื่อกดที่ชื่อผู้ป่วยแล้วจะเป็นหน้ารายการตรวจและรายการแพลงเมื่อนผู้ใช้ทั่วไป



รูปที่ 3.22 หน้าระบบการเพิ่มผู้ป่วยในการดูแลของพยาบาล

### 3.6.8 ระบบการยืนยันความถูกต้องของแพล

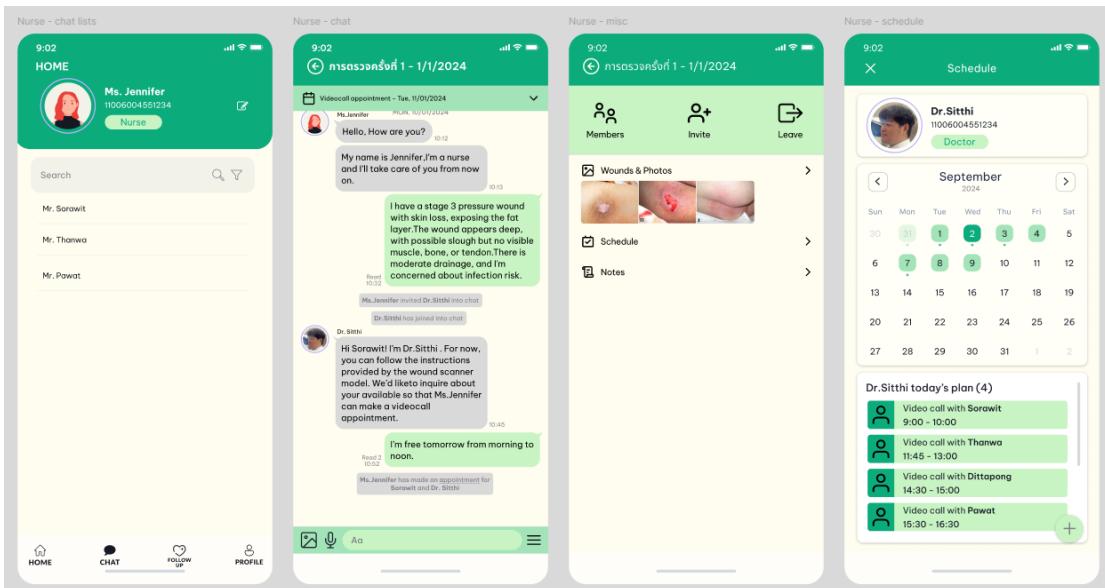
จัดทำหน้าระบบการยืนยันแพล โดยพยาบาลจะต้องเลือกผู้ป่วยในการดูแลที่ต้องการจะยืนยันแพลและกดเข้าไปที่รายการแพลเพื่อยืนยัน



รูปที่ 3.23 หน้าระบบการยืนยันความถูกต้องของแพล

### 3.6.9 ระบบแชทของพยาบาล

ได้จัดทำส่วนแชทของพยาบาล โดยพื้นฐานมีฟังก์ชันปกติเหมือนผู้ใช้งานทั่วไป เพิ่มเติมคือสามารถสร้างและแก้ไข note ได้



รูปที่ 3.24 หน้าระบบแชทของพยาบาล

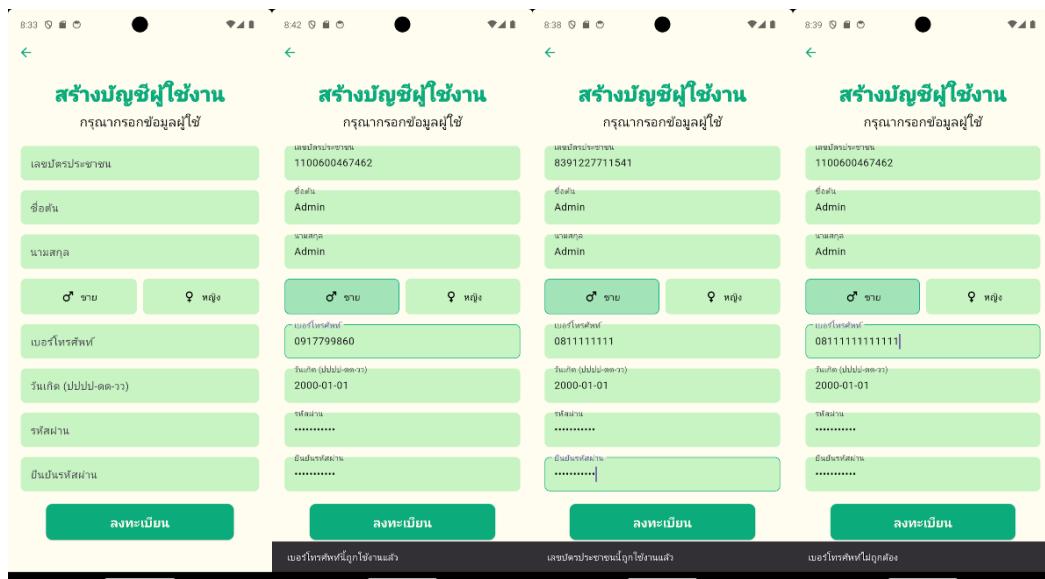
## บทที่ 4

### ผลการดำเนินงาน

#### 4.1 การพัฒนาและออกแบบในส่วนแอปพลิเคชัน

##### 4.1.1 ผลการทดลองการใช้ระบบลงทะเบียน (Registration)

เมื่อผู้ใช้ทำการกรอกข้อมูลเพื่อสร้างบัญชีใหม่ จะต้องระบุเลขประชาชน ชื่อ-นามสกุล เพศ เบอร์โทรศัพท์ วันเดือนปีเกิด รหัสผ่าน และยืนยันรหัสผ่าน หลังจากการกรอกข้อมูลครบถ้วนแล้ว ผู้ใช้สามารถทำการสร้างบัญชีได้ โดยระบบจะทำการตรวจสอบเลขรหัสบัตรประชาชนที่กรอกเข้ามา หากเลขรหัสบัตรประชาชนถูกต้อง ระบบจะนำผู้ใช้ไปยังหน้าเข้าสู่ระบบ (Login) เพื่อให้สามารถเข้าสู่บัญชีได้ แต่ในกรณีที่เกิดข้อผิดพลาด จะมีข้อความแสดงข้อผิดพลาด เพื่อแจ้งให้ผู้ใช้ทราบว่าข้อมูลที่กรอกไม่ถูกต้อง



รูปที่ 4.1 รูปผลการทดลองการใช้ระบบลงทะเบียน

##### 4.1.2 ผลการทดลองการเข้าสู่ระบบและออกจากระบบ (Login and Logout)

เมื่อผู้ใช้ทำการกรอกข้อมูลเพื่อเข้าสู่ระบบ จะต้องระบุเลขประชาชนและรหัสผ่าน หลังจากการกรอกข้อมูลครบถ้วนแล้ว ผู้ใช้สามารถทำการเข้าสู่ระบบได้ โดยระบบจะทำการตรวจสอบเลขรหัสบัตรประชาชนและรหัสผ่านที่กรอกเข้ามา หากข้อมูลถูกต้อง ระบบจะนำผู้ใช้ไปยังหน้าแสดงข้อมูลบัญชี แต่ในกรณีที่เลขรหัสบัตรประชาชนหรือรหัสผ่านไม่ตรงกับข้อมูลในระบบ หรือ เว้นว่าง จะมีข้อความแสดงข้อผิดพลาดขึ้น เพื่อแจ้งให้ผู้ใช้ทราบว่าข้อมูลที่กรอกไม่ถูกต้อง

เมื่อผู้ใช้ต้องการออกจากระบบ สามารถกดปุ่มออกจากระบบได้ ระบบจะทำการออกจากรับสัญญาณใช้และนำผู้ใช้กลับไปยังหน้าเข้าสู่ระบบ (Login) เพื่อให้ผู้ใช้สามารถเข้าสู่ระบบใหม่ได้

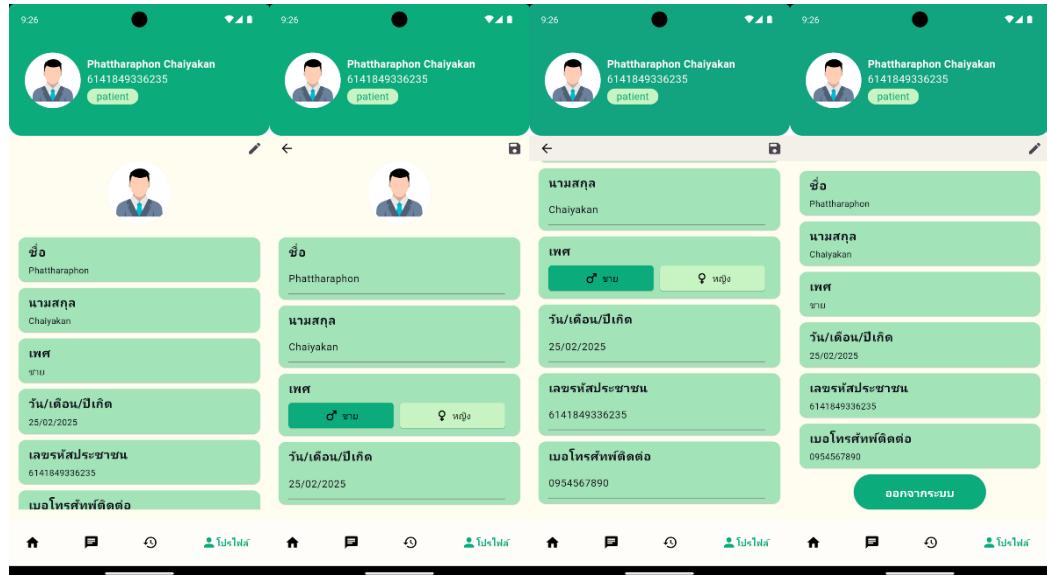


รูปที่ 4.2 รูปผลการทดลองการเข้าสู่ระบบ

#### 4.1.3 ผลการทดลองแสดงผลโปรไฟล์ (Profile)

เมื่อผู้ใช้เข้าสู่ระบบสำเร็จ ระบบจะทำการแสดงโปรไฟล์ของผู้ใช้บน Navbar ด้านบน โดยจะแสดงข้อมูลที่สำคัญ ได้แก่ ชื่อ นามสกุล เลขบัตรประชาชน รูปโปรไฟล์ และบทบาท (role) ของผู้ใช้ เพื่อให้ผู้ใช้สามารถตรวจสอบข้อมูลส่วนตัวได้อย่างรวดเร็วและสะดวกสบาย

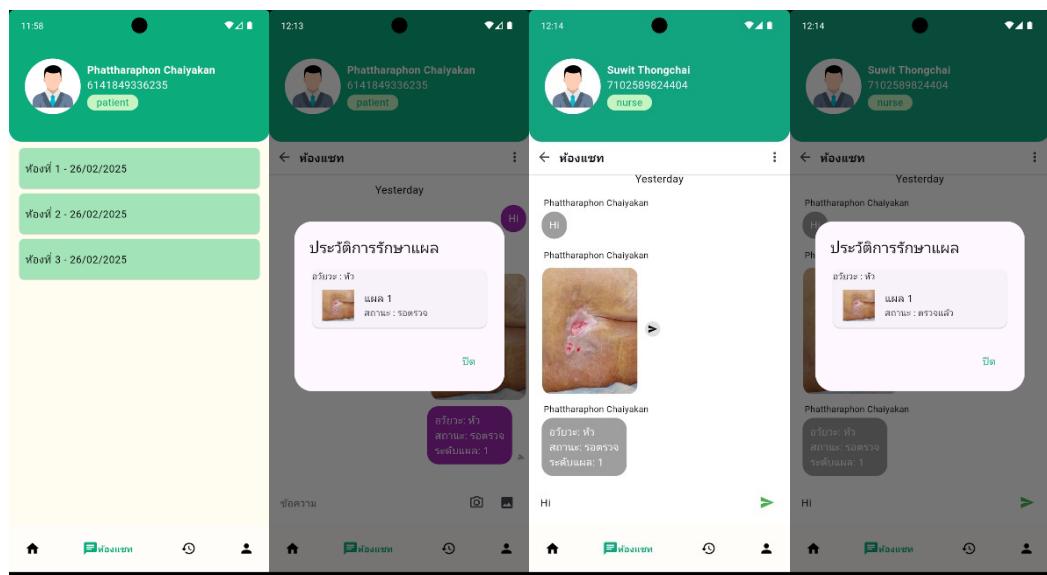
ผู้ใช้สามารถเข้าถึงหน้าแสดงโปรไฟล์โดยใช้ Bottom Nav ซึ่งจะมีตัวเลือกให้ผู้ใช้กดไปยังหน้า Show Profile โดยในหน้านี้จะแสดงข้อมูลส่วนตัวของผู้ใช้ย่างละเอียด ได้แก่ ชื่อ นามสกุล เพศ วันเดือนปีเกิด เลขบัตรประชาชน และเบอร์โทรศัพท์



รูปที่ 4.3 รูปผลการทดสอบแสดงผลໂປຣໄຟລ໌

#### 4.1.4 ผลการทดสอบเบื้องต้นของระบบแชท (Chat)

เมื่อผู้ใช้เข้าสู่หน้าแชท ระบบจะทำการแสดงจำนวนห้องแชทตามจำนวนครั้งของการตรวจ โดยแต่ละห้องแช法则มีการเก็บบันทึกนาฬิกา เมื่อทำการปิดแอปพลิเคชันและกลับมาเข้าห้องแชทอีกครั้งจะยังสามารถเห็นบันทึกนาฬิกาได้ และแชทสามารถรับส่งข้อความได้ real-time คือสามารถเห็นข้อความที่อีกฝ่ายส่งมาได้ทันที นอกจากนี้ระบบแชทสามารถส่งรูปแพลฟอร์มรายละเอียดจากการตรวจในครั้งนั้นๆเข้าสู่แชทได้

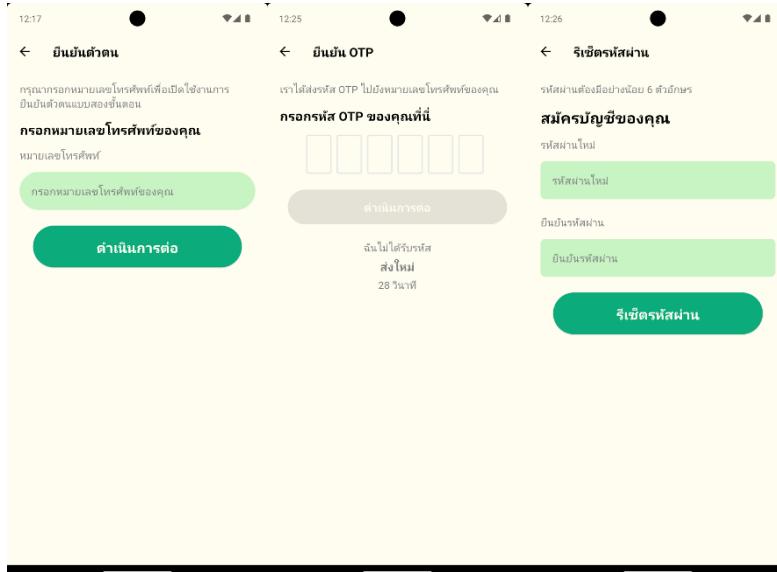


รูปที่ 4.4 รูปผลการทดสอบเบื้องต้นของระบบแชท

#### 4.1.5 ผลการทดลองการจัดการ OTP และการเปลี่ยนรหัสผ่าน (OTP and Change Password)

เมื่อผู้ใช้ทำการกรอกหมายเลขโทรศัพท์เพื่อรับ OTP ระบบจะทำการส่ง OTP ไปยังหมายเลขโทรศัพท์ที่ระบุ หากการส่ง OTP สำเร็จ ผู้ใช้จะได้รับข้อความ OTP และสามารถนำไปกรอกในช่องยืนยัน OTP ได้ หลังจากกรอก OTP ที่ได้รับ ระบบจะทำการตรวจสอบความถูกต้องของ OTP หาก OTP ถูกต้อง ระบบจะนำผู้ใช้ไปยังหน้าต่อไป แต่ในกรณีที่ OTP ไม่ถูกต้อง จะมีข้อความแสดงข้อผิดพลาดว่า “OTP ไม่ถูกต้อง กรุณาลองใหม่อีกครั้ง” เพื่อแจ้งให้ผู้ใช้ทราบว่า OTP ที่กรอกไม่ถูกต้อง

เมื่อผู้ใช้ต้องการเปลี่ยนรหัสผ่าน จะต้องกรอกรหัสผ่านใหม่และยืนยัน หากรหัสผ่านใหม่และยืนยันรหัสผ่านใหม่ตรงกัน ระบบจะทำการเปลี่ยนรหัสผ่านให้ผู้ใช้ แต่ในกรณีที่รหัสผ่านใหม่และยืนยันรหัสผ่านใหม่ไม่ตรงกัน จะมีข้อความแสดงข้อผิดพลาดว่า “รหัสผ่านไม่ตรงกัน” เพื่อแจ้งให้ผู้ใช้ทราบว่ารหัสผ่านที่กรอกไม่ตรงกัน

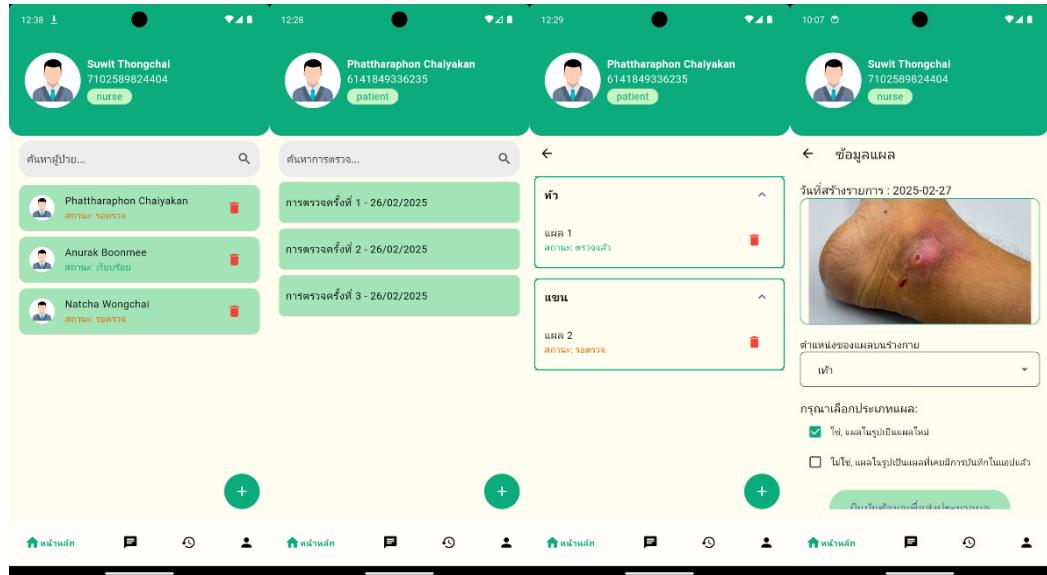


รูปที่ 4.5 รูปผลการทดลองการจัดการ OTP และการเปลี่ยนรหัสผ่าน

#### 4.1.6 ผลการทดลองการจัดการรายการตรวจและรายการแผล (Perusal and Wound)

เมื่อผู้ใช้เข้าสู่หน้ารายการตรวจ ระบบจะแสดงข้อมูลที่เกี่ยวข้องกับการตรวจและรายการแผลของผู้ป่วย โดยข้อมูลที่แสดงจะครอบคลุมรายละเอียดที่สำคัญ ดังนี้ รายชื่อผู้ป่วยและสถานะของแผล, รายการตรวจที่บันทึกไว้, รายละเอียดของรายการตรวจ, ปุ่มเพิ่มรายการตรวจ, การบันทึกข้อมูลรายการตรวจ และการลบรายการแผล

ระบบนี้ช่วยให้นักศึกษาทางการแพทย์สามารถติดตามและบริหารจัดการข้อมูลการตรวจและแผลของผู้ป่วยได้อย่างเป็นระบบ ลดข้อผิดพลาดในการบันทึกข้อมูล และช่วยให้สามารถให้การรักษาได้อย่างเหมาะสมและต่อเนื่อง

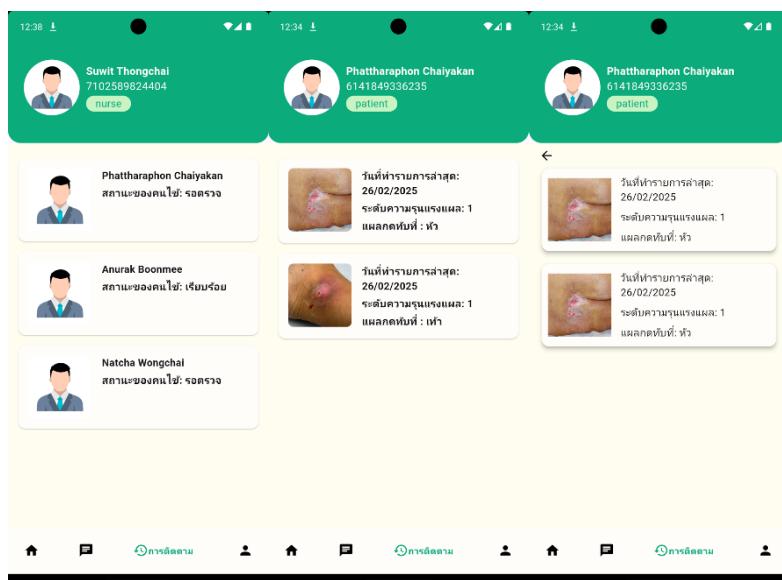


รูปที่ 4.6 รูปผลการทดสอบการจัดการรายการตรวจและรายการแผล

#### 4.1.7 ผลการทดสอบการแสดงผลการติดตามแผล (Wound Follow-Up)

เมื่อผู้ใช้เข้าสู่หน้าผลการติดตามแผล ระบบจะแสดงข้อมูลที่เกี่ยวข้องกับการติดตามแผลของผู้ป่วย โดยข้อมูลที่แสดงจะครอบคลุมรายละเอียดที่สำคัญ ดังนี้ รายชื่อผู้ป่วยและสถานะแผล, วันที่ทำการล่าสุด, ระดับความรุนแรงของแผล, ตำแหน่งของแผล, รูปภาพของแผล และปุ่มสำหรับดูรายละเอียดเพิ่มเติม

ระบบการติดตามแผลนี้ช่วยให้ทั้งบุคลากรทางการแพทย์และผู้ป่วยสามารถตรวจสอบและติดตามอาการของแผลได้อย่างมีประสิทธิภาพ ลดความเสี่ยงของภาวะแทรกซ้อน และช่วยให้สามารถวางแผนการรักษาได้ดียิ่งขึ้น

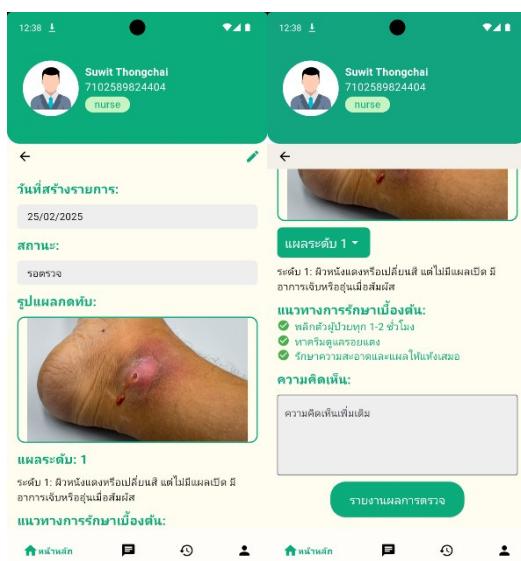


รูปที่ 4.7 รูปผลการทดสอบการแสดงผลการติดตามแผล

#### 4.1.8 ผลการทดลองการแสดงผลการวินิจฉัย (Diagnosis Result)

เมื่อผู้ใช้เข้าสู่หน้าผลการวินิจฉัย ระบบจะแสดงข้อมูลที่เกี่ยวข้องกับการวินิจฉัยแพลงของผู้ป่วย โดยข้อมูลที่แสดงจะครอบคลุมรายละเอียดที่สำคัญ ได้แก่ วันที่สร้างรายการ, สถานะของแพลง, ภาพแพลงของผู้ป่วย, ระดับความรุนแรงของแพลง, แนวทางการรักษาเบื้องต้น และความคิดเห็นเพิ่มเติม

ระบบนี้ช่วยให้บุคลากรทางการแพทย์สามารถติดตามและวิเคราะห์อาการของผู้ป่วยได้สะดวกยิ่งขึ้น โดยอำนวยความสะดวกในการจัดเก็บและเรียกดูข้อมูลการวินิจฉัยข้อนหลัง ลดข้อผิดพลาดในการบันทึกข้อมูล และเพิ่มประสิทธิภาพในการดูแลผู้ป่วย

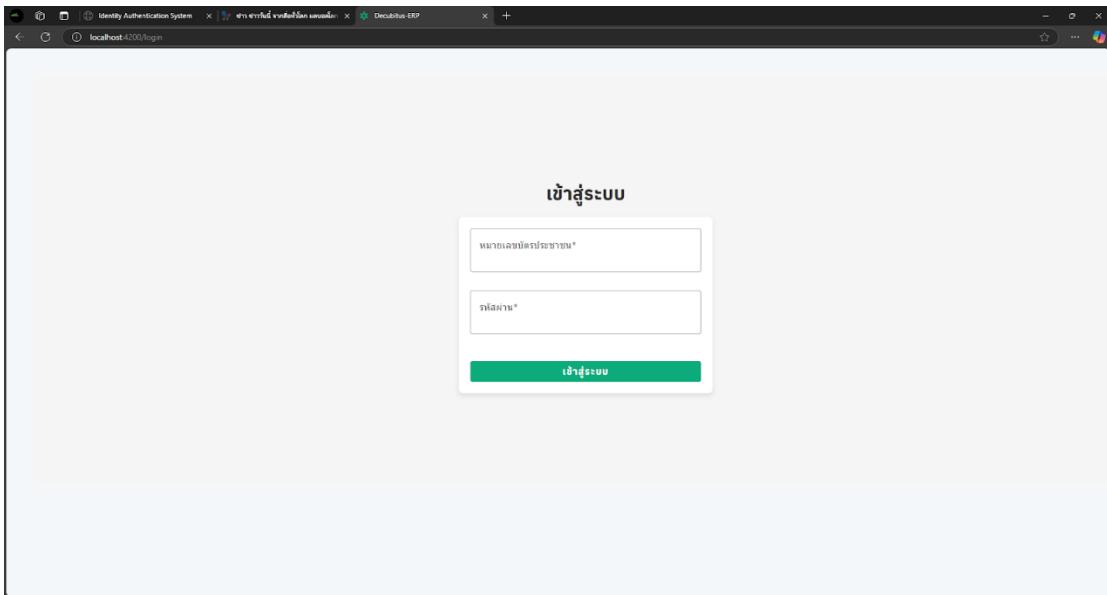


รูปที่ 4.8 รูปผลการทดลองการแสดงผลการวินิจฉัย

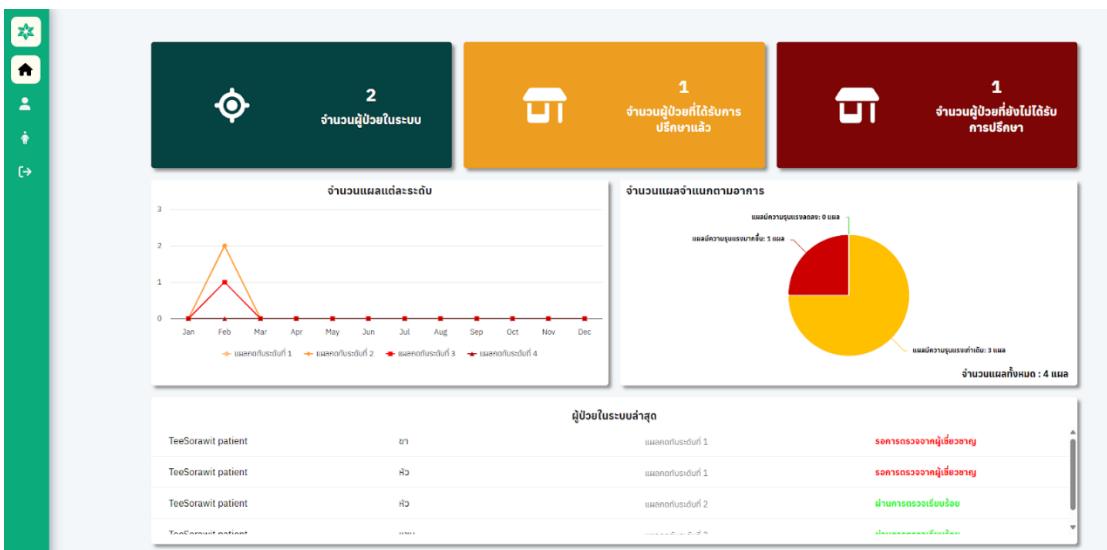
### 4.3 การพัฒนาระบบสำหรับผู้ดูแลระบบและแดชบอร์ด

#### 4.3.1 ระบบสำหรับผู้ดูแลระบบ

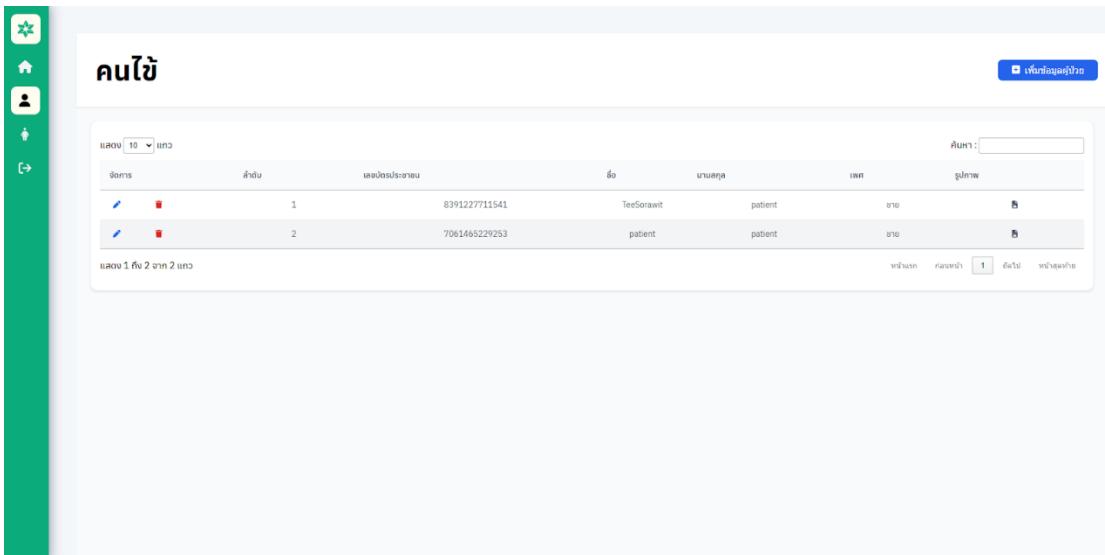
จัดทำส่วนการจัดการข้อมูลผู้ป่วยและพยาบาลของผู้ดูแลระบบรวมไปถึงหน้าแดชบอร์ดพร้อมทั้งเชื่อมต่อ API โดยที่ผู้ดูแลระบบจะสามารถเพิ่ม ลบ แก้ไขข้อมูล ของทั้งคนไข้และพยาบาลภายในระบบได้รวมถึงสามารถดูแดชบอร์ดข้อมูลผู้ป่วยได้



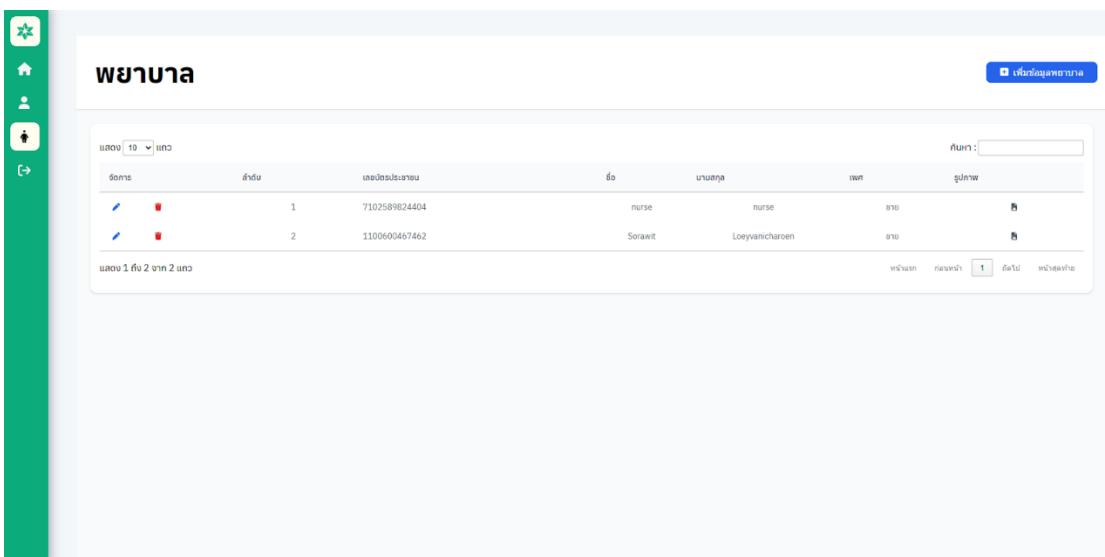
ຮູບທີ່ 4.9 ພໍມາເຂົ້າສູ່ຮະບບສໍາຫັນຜູ້ຄຸແລະບັນແປພາຍາລ



ຮູບທີ່ 4.10 ພໍມາແດ່ບນອ້ດສໍາຫັນຜູ້ຂໍ້ມູນແພລແລະຈຳນວນຜູ້ປ່າຍໃນຮະບບ



รูปที่ 4.11 หน้าของผู้ดูแลระบบที่ใช้ในการจัดการข้อมูลคนไข้



รูปที่ 4.12 หน้าของผู้ดูแลระบบที่ใช้ในการจัดการข้อมูลพยาบาล



The screenshot shows the 'Concierge' application interface. On the left is a vertical sidebar with icons for home, user profile, and navigation. The main area has a title 'คบไข้' (Concierge). Below it is a table with two rows, each containing a blue checkmark icon, a red cross icon, and the numbers '1' and '2'. A note at the bottom says 'เมื่อวันที่ 1 กันยายน 2562 จำนวน 2 คน'. To the right of the table is a large white dialog box titled 'เพิ่มข้อมูลผู้ป่วย' (Add New Record). Inside the dialog, there is a file upload icon, a button labeled 'ลืมรหัสผ่านรักษา', and four input fields: 'เลขประจำตัวประชาชน\*' (ID Number\*), 'เพศ\*' (Gender\*), 'เบอร์โทรศัพท์\*' (Phone Number\*), and 'ชื่อเด่น\*' (Name\*). At the top right of the dialog is a blue button labeled 'เพิ่มข้อมูลผู้ป่วย'.

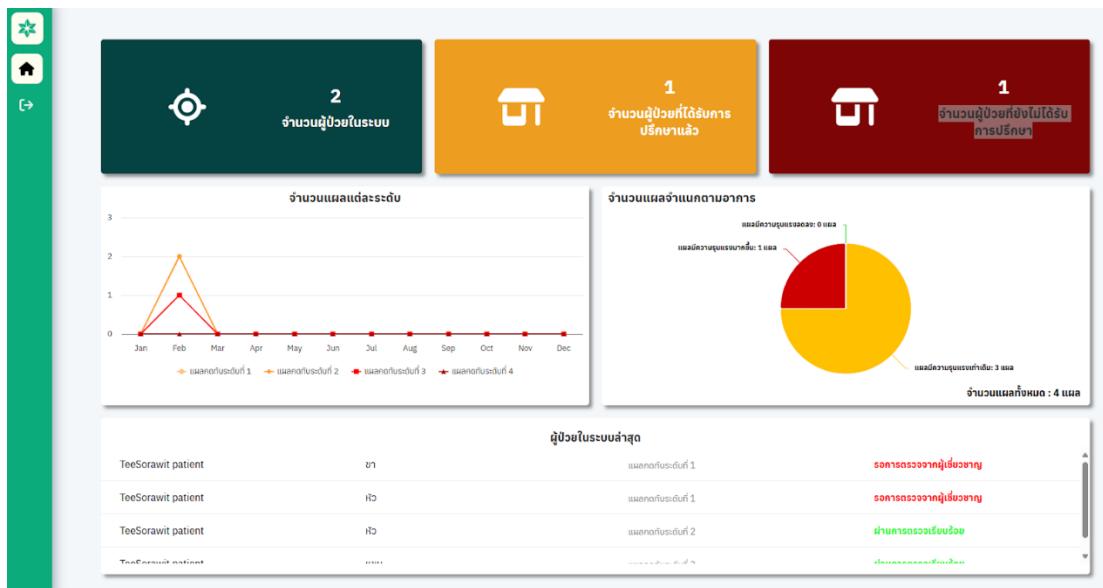
รูปที่ 4.15 หน้าของผู้ดูแลระบบที่ใช้ในการเพิ่มข้อมูลผู้ป่วย

The screenshot shows the 'Pharmacy' application interface. On the left is a vertical sidebar with icons for home, user profile, and navigation. The main area has a title 'พยาบาล' (Pharmacy). Below it is a table with two rows, each containing a blue checkmark icon, a red cross icon, and the numbers '1' and '2'. A note at the bottom says 'เมื่อวันที่ 1 กันยายน 2562 จำนวน 2 คน'. To the right of the table is a large white dialog box titled 'เพิ่มข้อมูลพยาบาล' (Add New Record). Inside the dialog, there is a file upload icon, a button labeled 'ลืมรหัสผ่านรักษา', and four input fields: 'เลขประจำตัวประชาชน\*' (ID Number\*), 'เพศ\*' (Gender\*), 'เบอร์โทรศัพท์\*' (Phone Number\*), and 'ชื่อเด่น\*' (Name\*). At the top right of the dialog is a blue button labeled 'เพิ่มข้อมูลพยาบาล'.

รูปที่ 4.16 หน้าของผู้ดูแลระบบที่ใช้ในการเพิ่มข้อมูลพยาบาล

#### 4.3.2 ระบบสำหรับพยาบาล

จัดทำส่วนแดชบอร์ดสำหรับพยาบาลเพื่อใช้ในการดูแลข้อมูลแพลกрайในระบบและเชื่อมต่อ API สำหรับแสดงข้อมูลในแดชบอร์ด โดยในหน้าแดชบอร์ดจะแสดงข้อมูลจำนวนผู้ป่วยภายในระบบ จำนวนผู้ป่วยที่ได้รับการปรึกษาแล้ว จำนวนผู้ป่วยที่ยังไม่ได้รับการปรึกษา จำนวนแพลตเติลระดับ จำนวนแพลตามอาการ และข้อมูลแพลในระบบล่าสุด

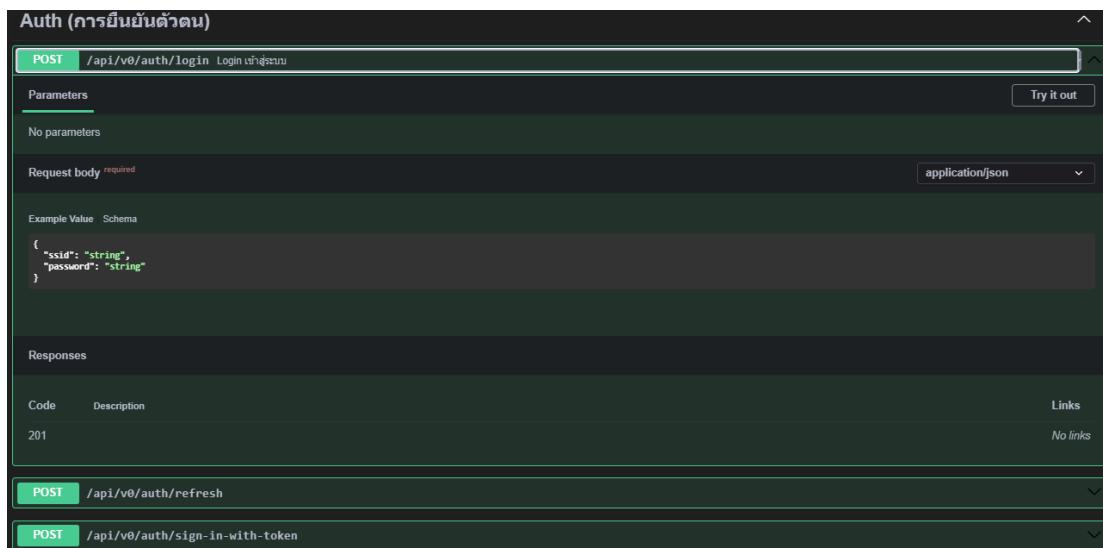


รูปที่ 4.17 หน้าของพยาบาลที่ใช้ในการดูข้อมูลผู้ป่วยในระบบ

### 4.3 การพัฒนาระบบในส่วน API

#### 4.3.1 API สำหรับระบบยืนยันตัวตน (Authentication API)

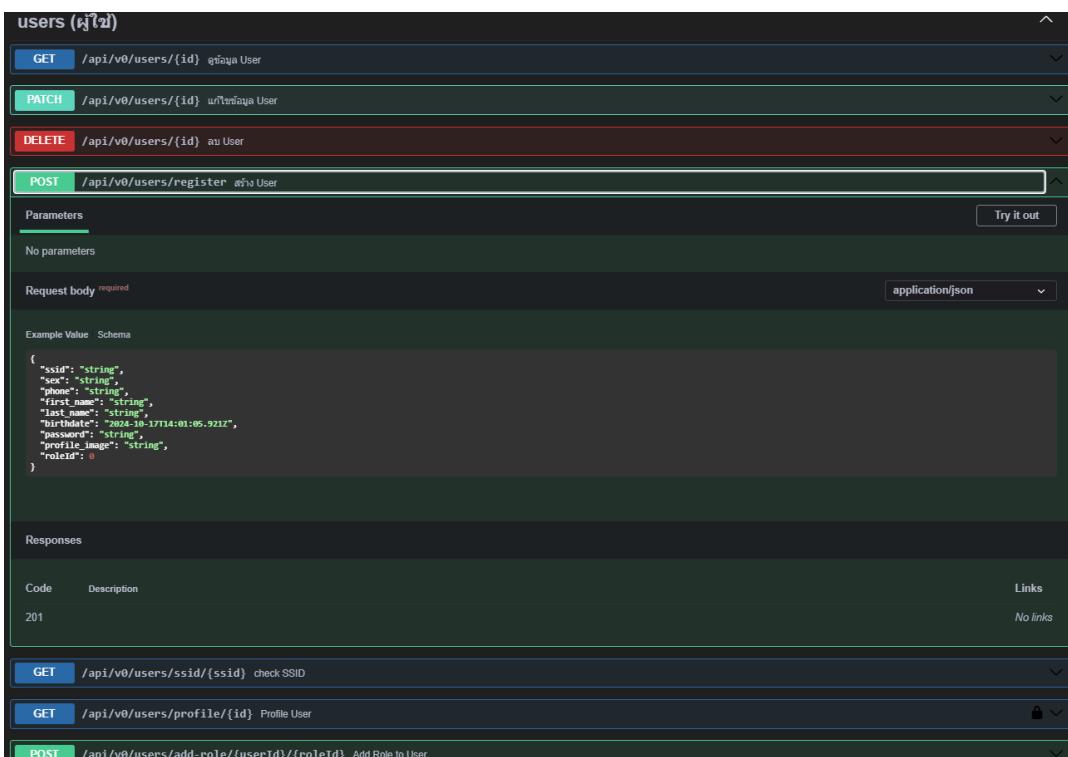
ผู้ใช้สามารถเข้าสู่ระบบได้โดยการกรอกเลขบัตรประชาชนและรหัสผ่านที่ลงทะเบียนไว้ ระบบจะทำการตรวจสอบข้อมูลที่ผู้ใช้กรอกกับฐานข้อมูลเพื่อยืนยันตัวตน หากข้อมูลถูกต้อง ระบบจะสร้าง Token ที่ใช้ในการเข้าถึงบริการต่างๆ ในระบบอย่างปลอดภัยและสะดวกสบาย นอกจากนี้ยังมีมาตรการรักษาความปลอดภัย เช่น การเข้ารหัสข้อมูล เพื่อป้องกันการเข้าถึงโดยไม่ได้รับอนุญาต การพัฒนา API นี้จึงเป็นส่วนสำคัญในการสร้างความเชื่อมั่นและความปลอดภัยให้กับผู้ใช้ในระบบ



รูปที่ 4.18 หน้า API สำหรับระบบยืนยันตัวตน

### 4.3.2 API สำหรับระบบการลงทะเบียน (Registration API)

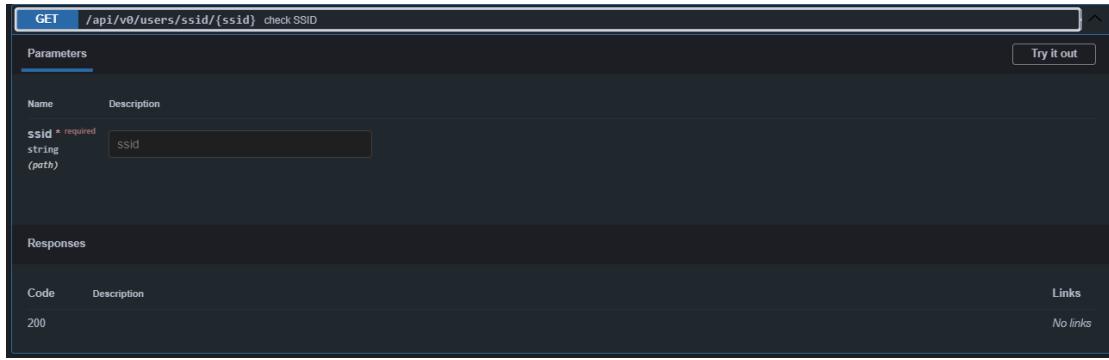
ในส่วนของการพัฒนาระบบ API สำหรับการลงทะเบียน ผู้ใช้สามารถสร้างบัญชีใหม่ได้โดยกรอกข้อมูลที่จำเป็น ได้แก่ เลขบัตรประชาชน เพศ ชื่อ นามสกุล วันเกิด หมายเลขโทรศัพท์ และรหัสผ่าน ระบบจะทำการตรวจสอบความถูกต้องของข้อมูลที่กรอกในฐานข้อมูล หากข้อมูลถูกต้อง ระบบจะทำการสร้างบัญชีผู้ใช้ใหม่ โดยยังต้องมีรหัสผ่านที่เข้ากัน ผู้ใช้สามารถเข้าสู่ระบบได้ทันทีและมีสิทธิ์ในการเพิ่มรายละเอียดอื่นๆ เช่น รูปโปรไฟล์ และข้อมูลเพิ่มเติมอื่นๆ ได้ในภายหลัง การพัฒนา API นี้ช่วยให้กระบวนการลงทะเบียนเป็นไปอย่างรวดเร็วและปลอดภัย พร้อมทั้งเปิดโอกาสให้ผู้ใช้สามารถปรับปรุงข้อมูลส่วนตัวได้ตามต้องการ



รูปที่ 4.19 หน้า API สำหรับระบบการลงทะเบียน

### 4.3.3 API สำหรับตรวจสอบเลขบัตรประชาชน (SSID Check API)

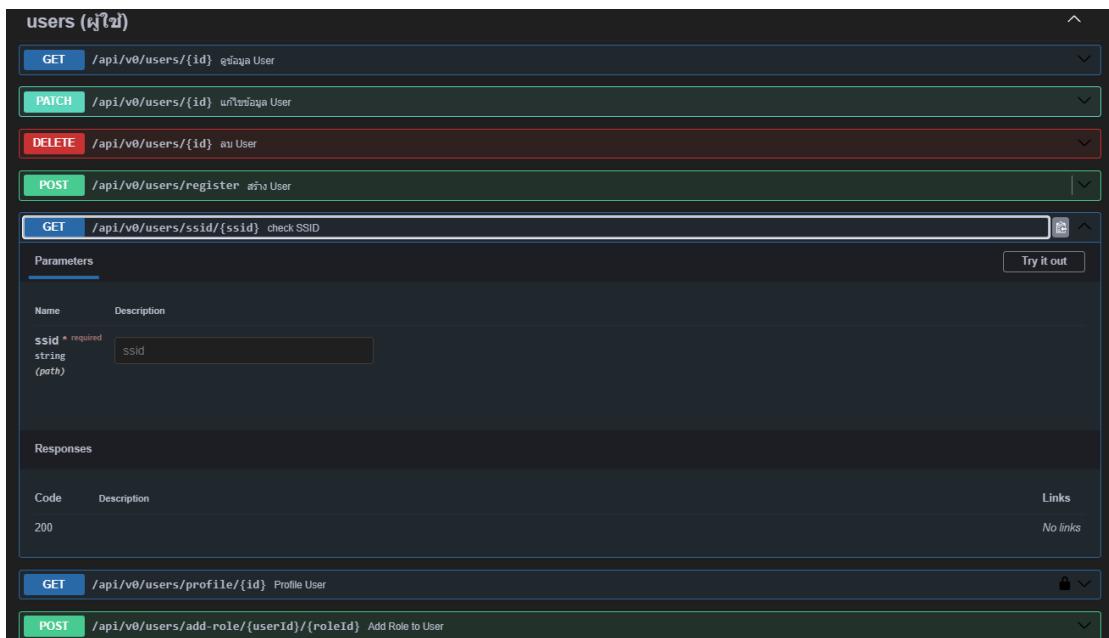
ในการพัฒนาระบบ API สำหรับตรวจสอบเลขบัตรประชาชน ผู้ใช้สามารถส่งคำขอเพื่อยืนยันความถูกต้องของเลขบัตรประชาชนที่กรอก โดยระบบจะทำการตรวจสอบข้อมูลในฐานข้อมูลเพื่อให้แน่ใจว่าเลขบัตรประชาชนดังกล่าวมีอยู่จริงและตรงตามข้อมูลที่ลงทะเบียนไว้ หากเลขบัตรประชาชนถูกต้อง ระบบจะส่งกลับข้อมูลสถานะการตรวจสอบ "ถูกต้อง" หรือ "ไม่ถูกต้อง" การพัฒนา API นี้มีวัตถุประสงค์เพื่อเพิ่มความสะดวกในการยืนยันตัวตนของผู้ใช้และลดความเสี่ยงจากการใช้ข้อมูลปลอมในการลงทะเบียนหรือเข้าสู่ระบบ



รูปที่ 4.20 หน้า API สำหรับตรวจสอบเลขบัตรประชาชน

#### 4.3.4 API สำหรับดึงข้อมูลโปรไฟล์ของผู้ใช้ (User Profile Retrieval API)

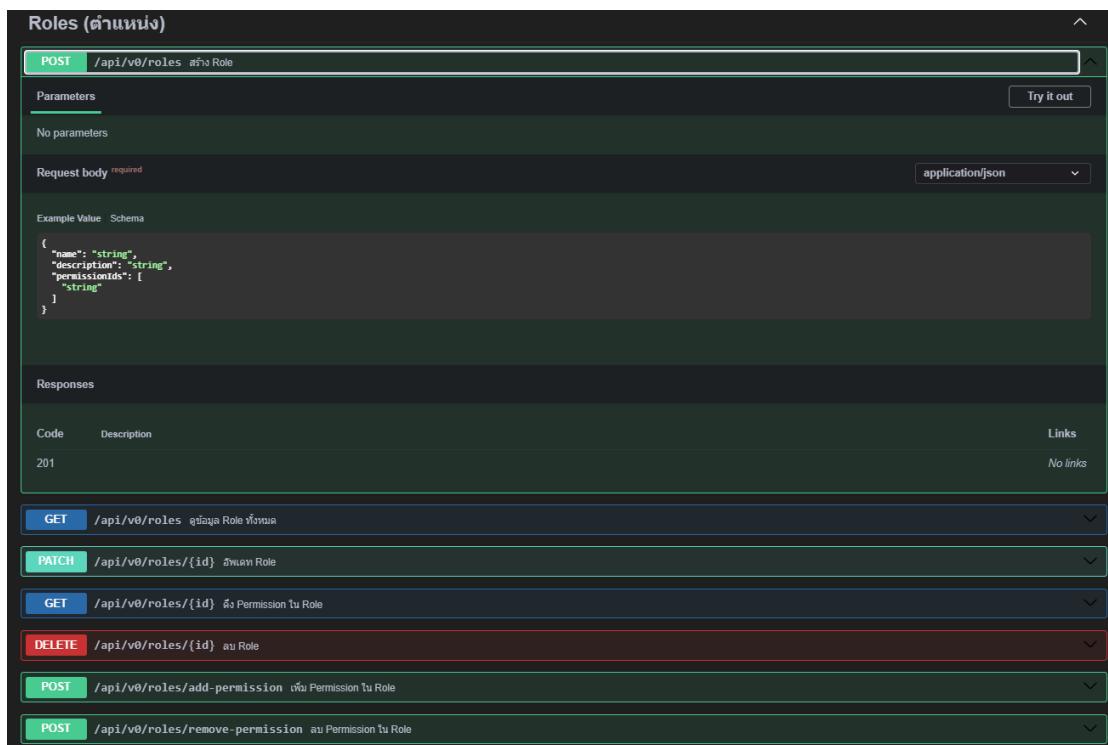
API นี้ถูกออกแบบมาเพื่อดึงข้อมูลโปรไฟล์ต่างๆ ของผู้ใช้จากฐานข้อมูล โดยต้องมีการตรวจสอบสิทธิ์ผ่านการใช้ access token ก่อนที่จะสามารถเข้าถึงข้อมูลได้ ผู้ใช้จะต้องส่งคำขอ GET พร้อมกับ access token ใน HTTP Authorization header ในรูปแบบ "Bearer" เมื่อระบบได้รับคำขอ และทำการตรวจสอบความถูกต้องของ access token และระบบจะทำการดึงข้อมูลโปรไฟล์ของผู้ใช้ ซึ่งอาจประกอบด้วยชื่อ, นามสกุล, อีเมล, วันเกิด และข้อมูลอื่นๆ ที่เกี่ยวข้อง จากนั้นจะส่งกลับข้อมูลในรูปแบบ JSON หากการตรวจสอบสิทธิ์สำเร็จ ข้อมูลโปรไฟล์จะถูกส่งกลับไปยังผู้ใช้ เพื่อให้สามารถใช้งานได้อย่างสะดวกและปลอดภัย



รูปที่ 4.21 หน้า API สำหรับดึงข้อมูลโปรไฟล์ของผู้ใช้

### 4.3.5 API ที่เกี่ยวกับ Roles (Roles API)

ในการพัฒนาระบบ API ที่เกี่ยวกับ Roles มีวัตถุประสงค์เพื่อจัดการสิทธิ์การเข้าถึงของผู้ใช้ในระบบ โดยสามารถสร้างบทบาท (Roles) ที่แตกต่างกันเพื่อกำหนด permissions ที่เฉพาะเจาะจงให้กับผู้ใช้แต่ละคน ได้อย่างมีประสิทธิภาพ



รูปที่ 4.22 หน้า API ที่เกี่ยวกับ Roles

### 4.3.6 API ที่เกี่ยวกับ Permissions (Permissions API)

API ถูกออกแบบมาเพื่อช่วยในการจัดการและควบคุมสิทธิ์การเข้าถึงไฟล์เอกสารต่างๆ ในระบบ โดยสามารถใช้ในการตรวจสอบสถานะของ permissions ที่ผู้ใช้ได้รับอนุญาตหรือไม่อนุญาตให้เข้าถึงไฟล์เอกสารต่างๆ ได้อย่างมีประสิทธิภาพ

Permissions (ສື່ອົງ)

**POST** /api/v0/permissions ສ້າງ Permission

Parameters

No parameters

Request body required

Example Value Schema

```
{
  "name": "string",
  "description": "string"
}
```

Responses

Code	Description	Links
201		No links

**PATCH** /api/v0/permissions/{id} ແກ້ໄຂຂໍ້ມູນ Permission

**DELETE** /api/v0/permissions/{id} ເລີຍຂໍ້ມູນ Permission

ຮູບທີ 4.23 ມ້າງ API ທີ່ເກີຍວັກນິກັນ Permissions

#### 4.3.7 API ສໍາຫັບການສ້າງຂໍ້ມູນການຕຽບ (Create Perusal API)

ໃນການພັດນາຮະບນ API ສໍາຫັບການສ້າງຂໍ້ມູນການຕຽບ ຜູ້ໃຊ້ສາມາດສ່າງຄຳຂອງເພື່ອບັນທຶກຂໍ້ມູນການຕຽບຈັກຢາ ໂດຍຂໍ້ມູນທີ່ຈຳເປັນຕົ້ນກຣອກ ໄດ້ແກ່ timestamp ຜົ່ງເປັນເວລາທີ່ຜູ້ໃຊ້ກັດສ່າງຄຳຂອງ ແລະ ID ຂອງຜູ້ໃຊ້ຈາກໃນຂະນັ້ນ ຮະບນຈະທຳການບັນທຶກຂໍ້ມູນແລ້ວນີ້ລັງໃນສູານຂໍ້ມູນເພື່ອໃຊ້ໃນການຕິດຕາມແລະວິເຄາະຫໍ່ຂໍ້ມູນການຕຽບໃນອນາຄາຕ

Perusal (ການຕຽບ)

**POST** /api/v0/perusal ສ້າງ ການຕຽບ

Parameters

No parameters

Request body required

Example Value Schema

```
{
  "perusal_date": "2024-10-17T14:22:39.424Z",
  "patient_id": 0
}
```

Responses

Code	Description	Links
201		No links

**PATCH** /api/v0/perusal/{id} ແກ້ໄຂ ການຕຽບ

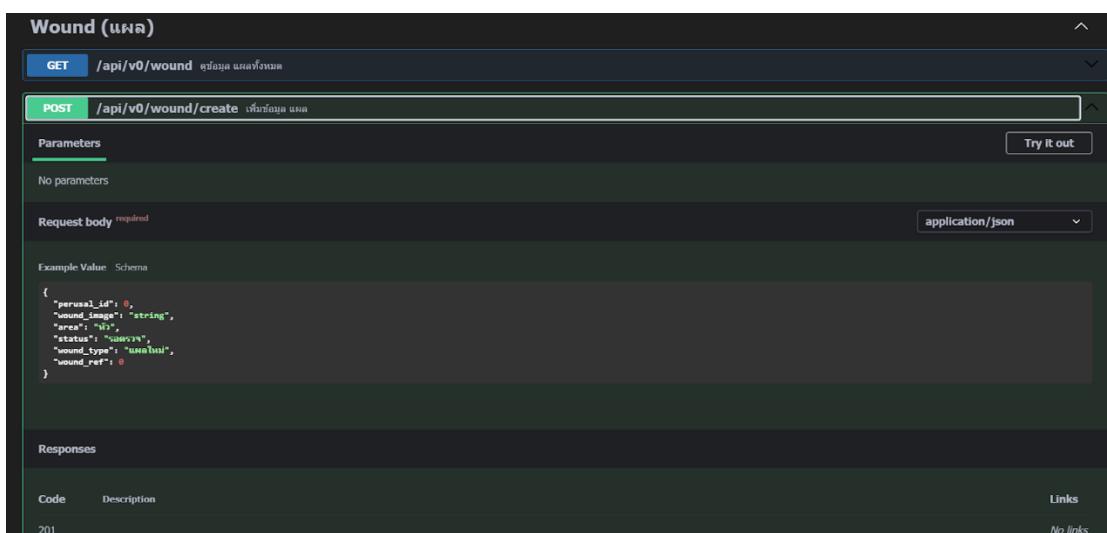
**DELETE** /api/v0/perusal/{id} ເລີຍ ການຕຽບ

**GET** /api/v0/perusal/Pages/{userId} ລົງຂໍ້ມູນການຕຽບ

ຮູບທີ 4.24 ມ້າງ API ສໍາຫັບການສ້າງຂໍ້ມູນການຕຽບ

#### 4.3.8 API สำหรับการเพิ่มข้อมูลแผล (Create Wound Information API)

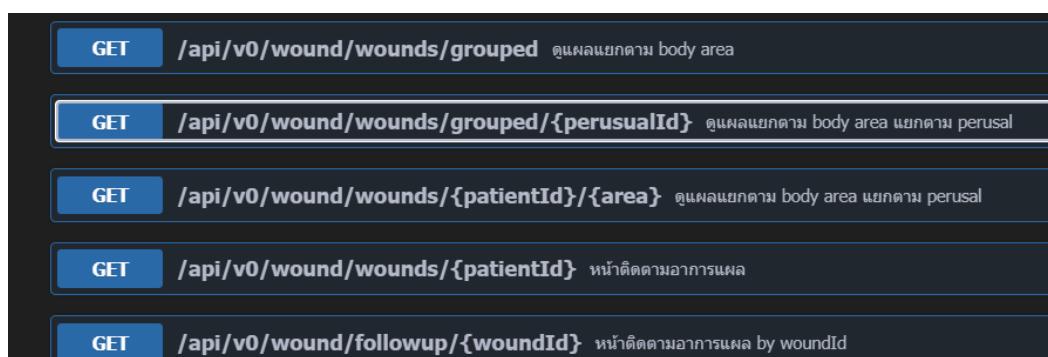
ในการพัฒนาระบบ API สำหรับการเพิ่มข้อมูลแผล ผู้ใช้สามารถส่งคำขอเพื่อบันทึกข้อมูลแผลใหม่ลงในฐานข้อมูลการตรวจหลังจากที่ได้สร้างข้อมูลการตรวจ (Create Perusal) แล้ว ข้อมูลที่จำเป็นต้องส่งประกอบด้วย หมายเลขการตรวจ (Perusal ID) ซึ่งระบุว่าข้อมูลแผลนี้เกี่ยวข้องกับการตรวจครั้งที่เท่าไหร่, รูปภาพแผล ที่ถ่ายในขณะนั้น, บริเวณที่ถ่ายแผล เช่น ตำแหน่งของร่างกายที่มีแผล และ สถานะ ของการตรวจว่า "ตรวจแล้ว" หรือ "รอตรวจ" ระบบจะทำการบันทึกข้อมูลเหล่านี้ลงในฐานข้อมูล พร้อมกับรายละเอียดของข้อมูลที่บันทึกไว้



รูปที่ 4.25 หน้า API สำหรับการเพิ่มข้อมูลแผล

#### 4.3.9 API สำหรับการแสดงข้อมูลแผล (Get Wound Information API)

API นี้รองรับการดึงข้อมูลแผลโดยสามารถจัดกลุ่มตามบริเวณร่างกาย แยกตาม Perusal ดูข้อมูลแผลของผู้ป่วยเฉพาะบริเวณหรือทั้งหมด รวมถึงคิดตามอาการแผลผ่านรหัสแผล



รูปที่ 4.26 หน้า API สำหรับการแสดงข้อมูลแผล

#### 4.3.10 API สำหรับการจัดการระดับความรุนแรงของแผล (Wound State Management API)

API นี้รองรับการจัดการข้อมูลระดับความรุนแรงของแผล โดยสามารถดึงข้อมูลทั้งหมด หรือเฉพาะรายการที่ต้องการ สร้าง แก้ไข และลบข้อมูล ได้ตามต้องการ นอกจากนี้ยังสามารถเพิ่มวิธีการรักษาให้สอดคล้องกับระดับความรุนแรงของแผล

<b>WoundState (ระดับความรุนแผล)</b>	
<b>GET</b>	/api/v0/woundstate ดึงข้อมูลทั้งหมด ระดับความรุนแผล
<b>POST</b>	/api/v0/woundstate สร้างข้อมูลระดับความรุนแผล
<b>GET</b>	/api/v0/woundstate/{id} ดึงข้อมูลเดียว ระดับความรุนแผล
<b>PATCH</b>	/api/v0/woundstate/{id} แก้ไขข้อมูลระดับความรุนแผล
<b>DELETE</b>	/api/v0/woundstate/{id} ลบข้อมูลระดับความรุนแผล
<b>POST</b>	/api/v0/woundstate/treat/{treatId}/{woundStateId} เพิ่มวิธีการรักษาตามระดับความรุนแผล

รูปที่ 4.27 หน้า API สำหรับการจัดการระดับความรุนแรงของแผล

#### 4.3.11 API สำหรับการจัดการข้อมูลการรักษา (Diagnosis Management API)

API นี้รองรับการจัดการข้อมูลการรักษา โดยสามารถดึงข้อมูลทั้งหมดหรือเฉพาะข้อมูลที่เกี่ยวข้องกับแพลทีระบุ สร้าง แก้ไข และลบข้อมูลการรักษาได้

<b>Diagnosis (การรักษา)</b>	
<b>GET</b>	/api/v0/diagnosis ดูข้อมูล การรักษาทั้งหมด
<b>POST</b>	/api/v0/diagnosis สร้างข้อมูล การรักษา
<b>PATCH</b>	/api/v0/diagnosis/{id} แก้ไขข้อมูล การรักษา
<b>DELETE</b>	/api/v0/diagnosis/{id} ลบข้อมูล การรักษา
<b>GET</b>	/api/v0/diagnosis/wound/{woundId} ดูข้อมูล การรักษา

รูปที่ 4.28 หน้า API สำหรับการจัดการข้อมูลการรักษา

#### 4.3.12 API สำหรับการจัดการห้อง (Room Management API)

API นี้รองรับการสร้างและจัดการห้อง โดยสามารถสร้างห้องทั่วไปหรือห้องสำหรับพยาบาล ดูรายการห้องทั้งหมด หรือดึงข้อมูลห้องที่ผู้ใช้เข้าร่วมได้ รวมถึงการเข้าห้องโดยตรงหรือผ่านการตรวจ และดูข้อมูลการตรวจของห้องที่ระบุ ช่วยให้การจัดการห้องและการตรวจรักษาทำได้อย่างเป็นระบบ

room (ห้อง)	
POST	/api/v0/rooms สร้างห้อง
GET	/api/v0/rooms ดูห้องทั้งหมด
POST	/api/v0/rooms/nurse สร้างห้อง
GET	/api/v0/rooms/join/{roomId}/{userId} เข้าห้อง
GET	/api/v0/rooms/perusal/{perusalId}/{userId} เข้าห้องโดยการตรวจ
GET	/api/v0/rooms/user/{userId} ห้องของผู้ใช้
GET	/api/v0/rooms/perusal/{roomId} ดูการตรวจ by roomId

รูปที่ 4.29 หน้า API สำหรับการจัดการห้อง

#### 4.3.13 API สำหรับการจัดการแชท (Chat Management API)

API นี้รองรับการส่งและดึงข้อความภายในห้องแชท โดยสามารถส่งข้อความใหม่และดึงประวัติการสนทนากำหนดของห้องที่ระบุ ช่วยให้การสื่อสารระหว่างผู้ใช้ภายในระบบ

chats (แชท)	
POST	/api/v0/chats ส่งข้อความ
GET	/api/v0/chats/{roomId} ดึงข้อความทั้งหมดของห้อง

รูปที่ 4.30 หน้า API สำหรับการจัดการแชท

#### 4.3.14 API สำหรับการจัดการการดูแล (Treatment Management API)

API นี้รองรับการสร้าง แก้ไข ลบ และดึงข้อมูลการดูแล โดยสามารถเรียกดูข้อมูลทั้งหมดหรือเฉพาะรายการที่ต้องการได้ ช่วยให้การจัดการแนวทางการดูแลผู้ป่วยเป็นไปอย่างเป็นระบบ

<b>treat (การดูแล)</b>	
<b>POST</b>	/api/v0/treat สร้าง การดูแล
<b>GET</b>	/api/v0/treat ตั้งชื่อ模 การดูแลทั้งหมด
<b>PATCH</b>	/api/v0/treat/{id} แก้ไข การดูแล
<b>DELETE</b>	/api/v0/treat/{id} ลบ การดูแล
<b>GET</b>	/api/v0/treat/{id} ตั้งชื่อ模 การดูแล

รูปที่ 4.31 หน้า API สำหรับการจัดการการดูแล

#### 4.3.15 API สำหรับแดชบอร์ด (Dashboard API)

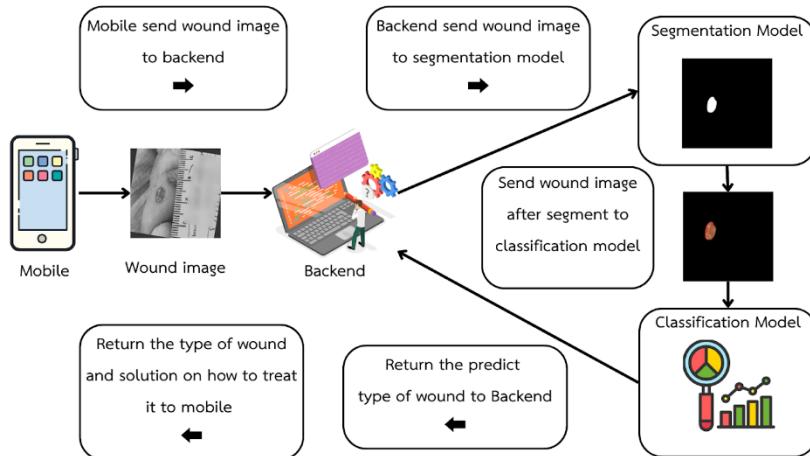
API นี้ใช้สำหรับดึงข้อมูลวิเคราะห์ต่างๆ บนแดชบอร์ดของระบบ ERP โดยแยกเป็นส่วนต่างๆ เพื่อแสดงข้อมูลสำคัญและช่วยให้ผู้ใช้สามารถวิเคราะห์และติดตามสถานะของระบบได้อย่างมีประสิทธิภาพ

<b>dashboard (แดชบอร์ด)</b>	
<b>GET</b>	/api/v0/dashboard/topwidget Dashboard Topwidget
<b>GET</b>	/api/v0/dashboard/middlewidget Dashboard Middlewidget
<b>GET</b>	/api/v0/dashboard/middlewidget2 Dashboard Middlewidget2
<b>GET</b>	/api/v0/dashboard/bottomwidget Dashboard Bottomwidget

รูปที่ 4.32 หน้า API สำหรับแดชบอร์ด

## 4.4 การพัฒนาระบบในส่วนของโมเดล

### 4.4.1 ภาพรวมการทำงานของระบบประมวลผลภาพแพลตฟอร์ม



รูปที่ 4.33 รูปภาพรวมการทำงานของระบบในส่วนโมเดล

ระบบนี้เริ่มจากผู้ใช้ถ่ายภาพแพลตฟอร์มอุปกรณ์มือถือ และส่งภาพไปยังระบบหลังบ้าน (Backend) ซึ่งจะประมวลผลโดยส่งภาพแพลตฟอร์มเข้าสู่โมเดลแบ่งส่วนภาพ (Segmentation Model) เพื่อระบุตำแหน่งของแผล จากนั้นภาพที่ผ่านการแบ่งส่วนจะถูกส่งไปยังโมเดลจำแนกประเภท (Classification Model) เพื่อวิเคราะห์และจำแนกประเภทของแพลตฟอร์ม เมื่อได้รับผลการวิเคราะห์ระบบจะถ่ายข้อมูลประเภทของแพลตฟอร์มและคำแนะนำการรักษาลับไปยังอุปกรณ์มือถือของผู้ใช้

### 4.4.2 ฮาร์ดแวร์และซอฟต์แวร์

สำหรับการประมวลผลและฝึกสอนโมเดล เราใช้บริการ GPU จาก Kaggle และ Google Colab ซึ่งรองรับการใช้งาน CUDA เพื่อเร่งความเร็วในการประมวลผลโมเดลอย่างมีประสิทธิภาพ รายละเอียดการเปรียบเทียบทรัพยากรองแต่ละบริการมีดังต่อไปนี้

ตารางที่ 4.1 ตารางเปรียบเทียบทรัพยากรของแต่ละบริการ

คุณสมบัติ	Kaggle	Google Colab
ประเภท GPU	P100 GPU	T4 GPU
RAM	16 GB	15 GB
การใช้งาน CUDA	รองรับ	รองรับ
ระยะเวลาการใช้งาน	30 ชั่วโมงต่อสัปดาห์	2 ชั่วโมง 30นาทีต่อวัน
ค่าใช้จ่าย	ฟรี / มีค่าใช้จ่าย	ฟรี / มีค่าใช้จ่าย

#### 4.4.3 กระบวนการพัฒนาโมเดลสำหรับการแบ่งส่วนภาพ (Segmentation Model)

##### 4.4.3.1 การเปรียบเทียบประสิทธิภาพของโมเดลสำหรับการแบ่งส่วนภาพ (Segmentation Model)

ในการทดสอบนี้ เราได้ทำการเปรียบเทียบประสิทธิภาพของโมเดลแบ่งส่วนภาพจำนวนสองโมเดล ได้แก่ U-Net และ MobileNetV2 ในการแบ่งส่วนภาพแพลง โดยใช้เมตริกที่นิยมใช้ในการประเมินประสิทธิภาพการแบ่งส่วนภาพ ได้แก่ ค่า Intersection over Union (IoU) และ ค่า Dice Coefficient (Dice) ซึ่งถือเป็นเมตริกที่สำคัญในการประเมินความแม่นยำของการแบ่งส่วนภาพ

ทั้งสองโมเดลจะถูกเปรียบเทียบเพื่อระบุว่าโมเดลใดมีประสิทธิภาพที่ดีกว่าในการแบ่งส่วนภาพแพลงเพื่อการใช้งานในอนาคต โดยในขั้นตอนการพัฒนาโมเดลสำหรับการแบ่งส่วนภาพ (Segmentation Model) จะถูกแบ่งออกเป็นสองช่วงการทดลอง ได้แก่

การทดลองที่ 1: เปรียบเทียบประสิทธิภาพของโมเดลที่นำมาใช้ เพื่อวิเคราะห์ความแตกต่างในประสิทธิภาพของแต่ละโมเดล

การทดลองที่ 2: เมื่อรับ��ได้ว่าโมเดลใดมีประสิทธิภาพที่ดีกว่า จะทำการปรับพารามิเตอร์และจำนวนชุดข้อมูลเพื่อเปรียบเทียบผลลัพธ์ของโมเดลนั้นอย่างละเอียด

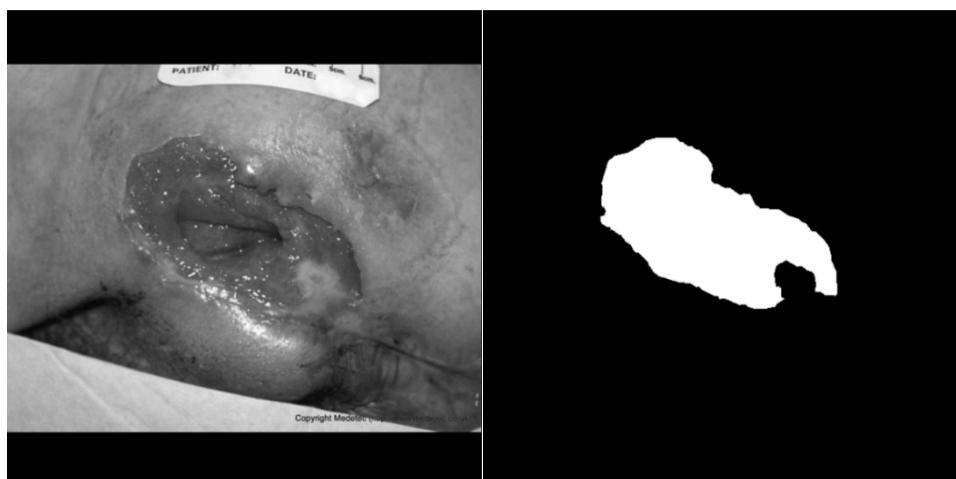
##### 4.4.3.2 สภาพแวดล้อมในการทดลอง

###### 4.4.3.2.1 ชุดข้อมูล (Dataset) สำหรับการทดลองครั้งที่ 1

ใช้ชุดข้อมูล "Wound images segmentation" จาก Kaggle ประกอบด้วยภาพแพลงทั้งหมด 2,760 ภาพ โดยที่จะประกอบด้วยชุดข้อมูลที่เป็นรูปแพลงและชุดข้อมูลที่เป็นหน้ากากของแพลง โดยชุดข้อมูลมีรายละเอียดดังนี้ดังนี้

ตารางที่ 4.2 ตารางแสดงรายละเอียดของชุดข้อมูลที่นำมาใช้ในการพัฒนาโมเดลสำหรับการแบ่งส่วนภาพ (Segmentation Model) สำหรับการทดลองครั้งที่ 1

ชุดข้อมูล	จำนวน
train_images	2208 รูป
train_masks	2208 รูป
test_images	552 รูป
test_masks	552 รูป



รูปที่ 4.34 ตัวอย่างรูปภาพแพลและหน้ากากของแพลสำหรับการทดลองครั้งที่ 1

#### 4.4.3.2.2 ชุดข้อมูล (Dataset) สำหรับการทดลองครั้งที่ 2

ในการทดลองครั้งที่ 2 ได้ใช้ชุดข้อมูลเดียวกันกับที่ใช้ในการทดลองครั้งที่ 1 แต่มีการเพิ่มขั้นตอน การเพิ่มข้อมูล (Augmentation) ในส่วนของชุด train\_images และ train\_masks เพื่อเพิ่มจำนวนข้อมูลที่ใช้ในการฝึกสอน โมเดล รายละเอียดของชุดข้อมูลมีดังนี้

ตารางที่ 4.3 ตารางแสดงรายละเอียดของชุดข้อมูลที่นำมาใช้ในการพัฒนาโมเดลสำหรับการแบ่งส่วนภาพ (Segmentation Model) สำหรับการทดลองครั้งที่ 2

ชุดข้อมูล	จำนวน
train_images	4416 รูป
train_masks	4416 รูป
test_images	552 รูป
test_masks	552 รูป

โดยในส่วนของกระบวนการเพิ่มข้อมูล (Augmentation) มีรายละเอียดขั้นตอนดังนี้

#### การสุ่มกลับด้านภาพแนวนอน (Random Horizontal Flip)

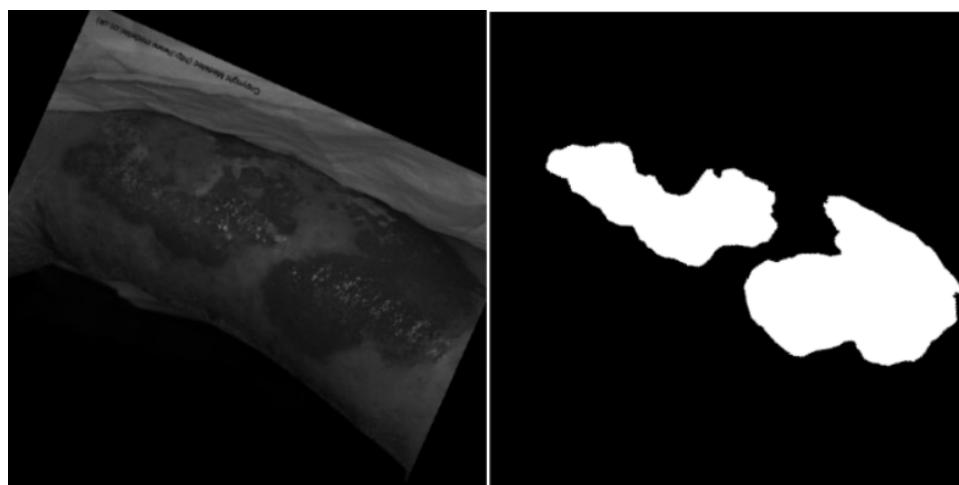
มีการสุ่มกลับด้านภาพและมาสก์ในแนวตั้ง โดยมีโอกาส 50% ที่จะกลับด้าน เพื่อลดการพึ่งพาตำแหน่งทางแนวตั้งของวัตถุในภาพ

#### การสุ่มหมุนภาพ (Random Rotation)

ภาพและมาสก์จะถูกหมุนในช่วงมุมระหว่าง -30 ถึง 30 องศา

#### การสุ่มปรับความสว่างและสี (Color Jitter)

ใช้การสุ่มปรับค่าความสว่าง, ความเปรียบต่าง, ความอิมตัวของสี และโทนสี (Hue) ของภาพ โดยมีโอกาส 40% ที่จะทำการปรับค่าเหล่านี้ แต่การปรับนี้จะไม่ถูกใช้กับมาสก์เพื่อรักษาคุณสมบัติของมาสก์เดิมไว้



รูปที่ 4.35 ตัวอย่างรูปภาพแพลและหน้ากากของแพลสำหรับการทดลองครั้งที่ 2

#### 4.4.3.2.3 การตั้งค่าพารามิเตอร์ในโมเดลสำหรับการทดลองครั้งที่ 1

ตารางที่ 4.4 ค่าพารามิเตอร์ที่ใช้ในการเทรนโมเดลในการทดลองรูปแบบที่ 1

พารามิเตอร์	ค่าพารามิเตอร์
number_of_dataset	2206 images
number_of_test_dataset	number_of_test_dataset
segmentation_model	U-net, Mobilenetv2
image_size	512x512
num_epochs	10
augment_factor	1
learning_rate	0.001
batch_size	4

#### 4.4.3.2.4 การตั้งค่าพารามิเตอร์ในโมเดลสำหรับการทดลองครั้งที่ 2

ตารางที่ 4.5 ค่าพารามิเตอร์ที่ใช้ในการเทรนโมเดลในการทดลองรูปแบบที่ 1

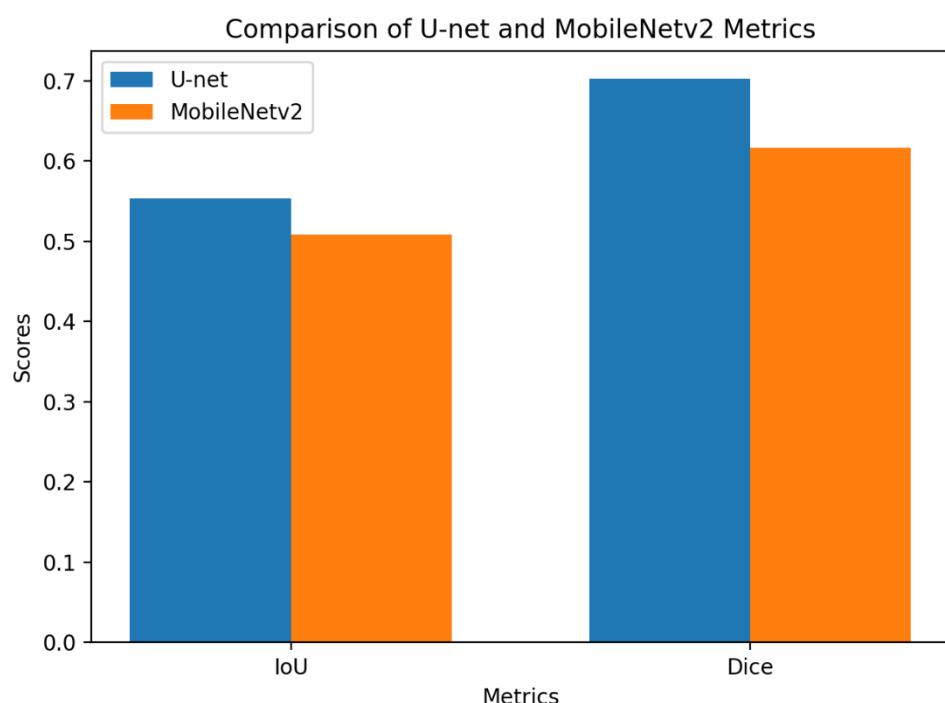
พารามิเตอร์	ค่าพารามิเตอร์
number_of_dataset	4412 images
number_of_test_dataset	number_of_test_dataset
segmentation_model	U-net
image_size	512x512
num_epochs	10,30,50
augment_factor	2
learning_rate	0.001
batch_size	4

#### 4.4.3.3 ผลการทดลองรูปแบบที่ 1 ระหว่าง U-net และ MobileNetV2

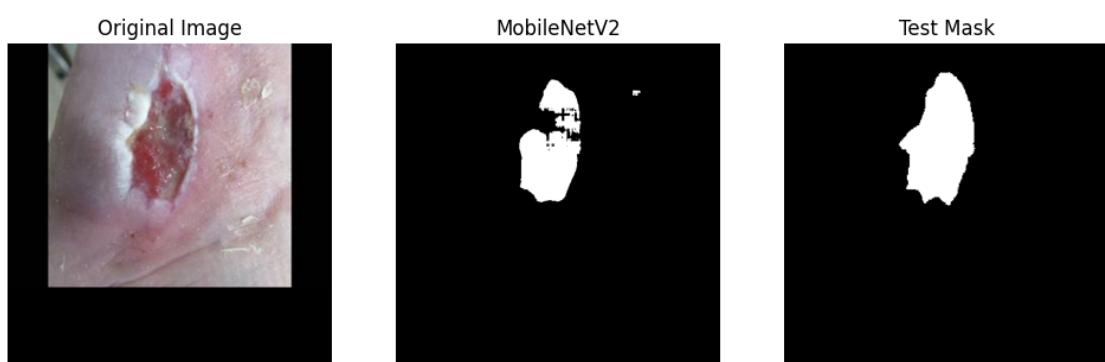
ในการทดลองนี้ ได้นำผลลัพธ์จากการทำนาย (predict\_mask) ของโมเดลทั้งสองมาเปรียบเทียบกับรูปจริงของ mask ทดสอบ (test\_mask) โดยใช้เมทริก Intersection over Union (IoU) และ Dice Coefficient (Dice) เพื่อประเมินความแม่นยำและคุณภาพของการทำนายจากรูปทดสอบ (test\_images) 552 รูป ผลการเปรียบเทียบแสดงดังนี้

ตารางที่ 4.6 ตารางวัดผลค่าเมترิกของการทดลองที่ 1

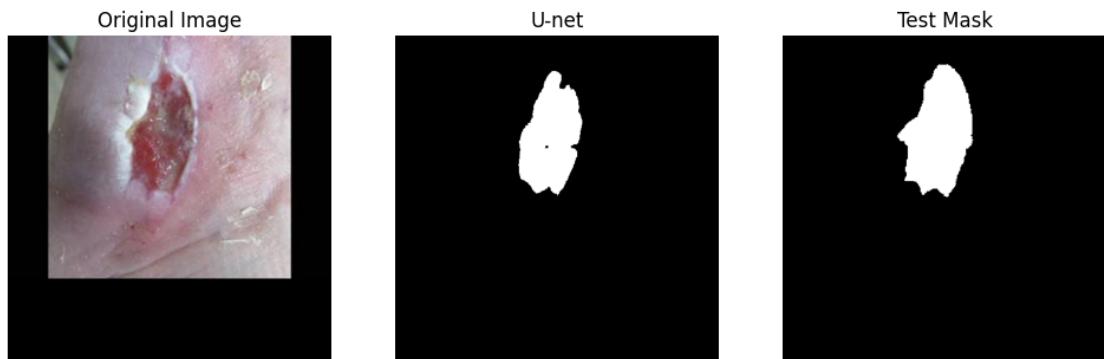
เมต릭	U-net	Mobilenetv2
IoU	0.5534	0.5082
Dice	0.7024	0.6164



รูปที่ 4.36 กราฟแท่งเปรียบค่าเมตริกของการทดลองที่ 1



รูปที่ 4.37 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ MobileNetV2 กับ รูปทดสอบ



รูปที่ 4.38 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ MobileNetV2 กับ รูปทดสอบ

จากการทดลองรูปแบบที่ 1 ใช้ neural network U-net และ MobileNetV2 มาทำการทำนายผล (predict\_mask) และเปรียบเทียบกับข้อมูลจริง (test\_mask) โดยใช้เมตริก IoU และ Dice Coefficient เพื่อประเมินความแม่นยำของการทำนาย พบว่า โมเดล U-net ให้ผลลัพธ์ที่มีความแม่นยำสูงกว่า MobileNetV2 ทั้งในด้านค่า IoU และ Dice Coefficient โดยค่า IoU ของ U-net อยู่ที่ 0.5534 ในขณะที่ MobileNetV2 มีค่า IoU อยู่ที่ 0.5082 ส่วนค่า Dice ของ U-net อยู่ที่ 0.7024 ซึ่งสูงกว่า MobileNetV2 ที่มีค่า Dice เพียง 0.6164

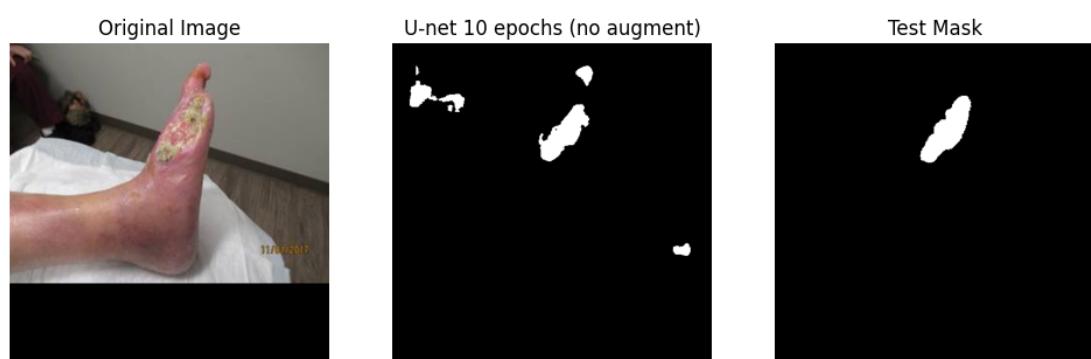
จากการทดลองแสดงให้เห็นว่า U-net มีประสิทธิภาพดีกว่าในการทำงานกับข้อมูลชุดนี้ ทั้งในเรื่องความสอดคล้องระหว่าง mask ที่ทำนายและ mask จริง โดยเฉพาะอย่างยิ่ง เมตริก Dice ซึ่งแสดงถึงความแม่นยำในการทำนายขอบเขตของวัตถุที่ชัดเจน

#### 4.4.3.4 ผลการทดลองรูปแบบที่ 2 เปรียบเทียบการแบ่งส่วนภาพโดยใช้การเพิ่มข้อมูล (Data Augmentation) ด้วย U-Net เทียบกับการไม่เพิ่มข้อมูล

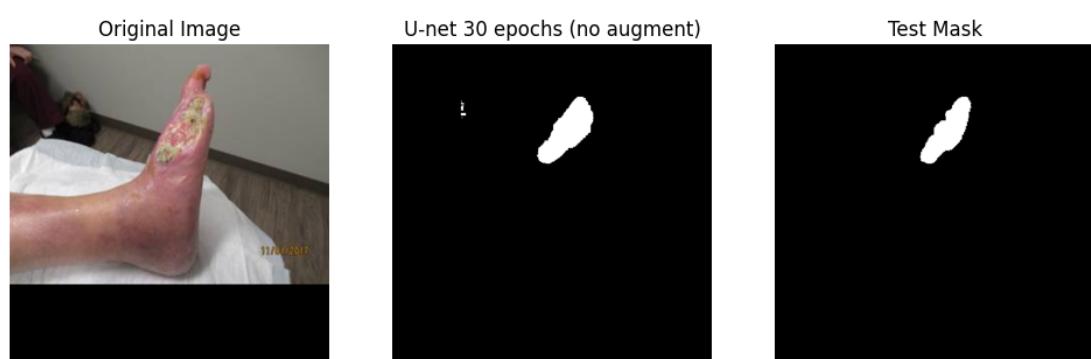
ในการทดลองนี้ ได้นำผลลัพธ์จากการทำนาย (predict\_mask) ของ U-net โมเดลที่ฝึกสอนโดยใช้การเพิ่มข้อมูลมาเปรียบเทียบกับที่ฝึกสอนโดยไม่เพิ่มข้อมูลแล้วนำมาเปรียบเทียบกับรูปจริงของ mask ทดสอบ (test\_mask) โดยใช้เมตริก Intersection over Union (IoU) และ Dice Coefficient (Dice) เพื่อประเมินความแม่นยำเฉลี่ยของการทำนายจากรูปทดสอบ(test\_images) 552 รูป ผลการเปรียบเทียบแสดงดังนี้

ตารางที่ 4.7 ตารางวัดผลค่าเมทริกของการทดลองที่ 2

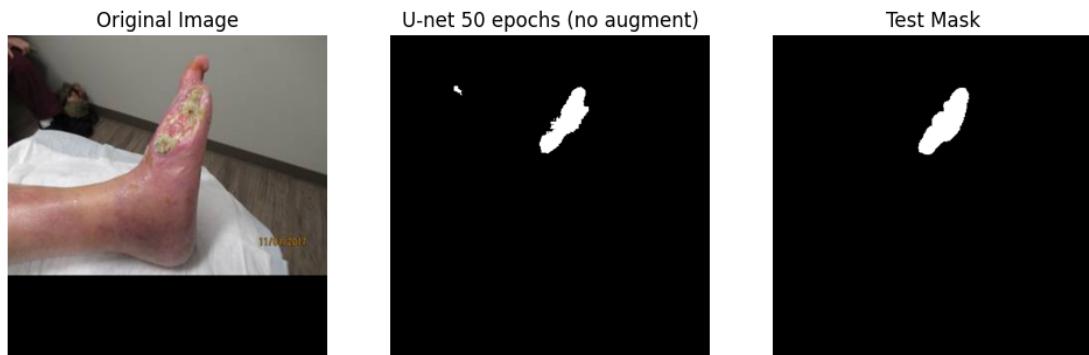
รอบ epochs	IoU (ไม่เพิ่มข้อมูล)	IoU (เพิ่มข้อมูล)	Dice (ไม่เพิ่มข้อมูล)	Dice (เพิ่มข้อมูล)
ค่าเฉลี่ยรวม (Average)	0.6226	0.6500	0.7542	0.7775
10 epochs	0.5294	0.5879	0.6754	0.7281
30 epochs	0.6571	0.6434	0.7843	0.7742
50 epochs	0.6814	0.7189	0.8030	0.8303



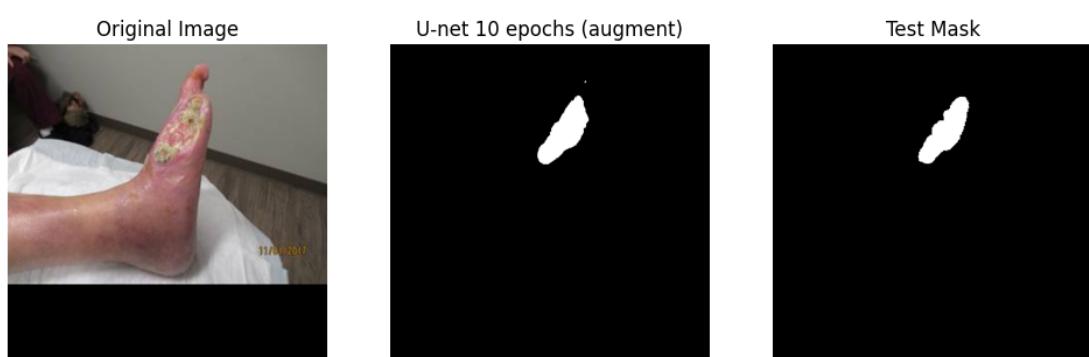
รูปที่ 4.39 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-net 10 epochs ที่ไม่มีการเพิ่มจำนวน กับ รูปทดสอบ



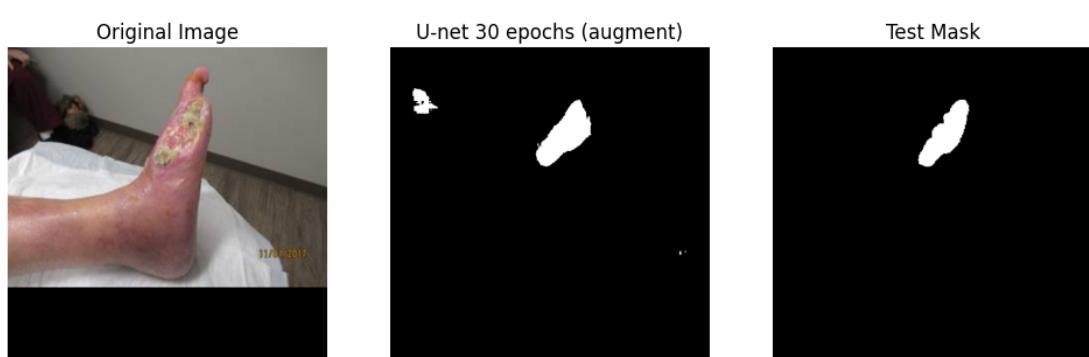
รูปที่ 4.40 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-net 30 epochs ที่ไม่มีการเพิ่มจำนวน กับ รูปทดสอบ



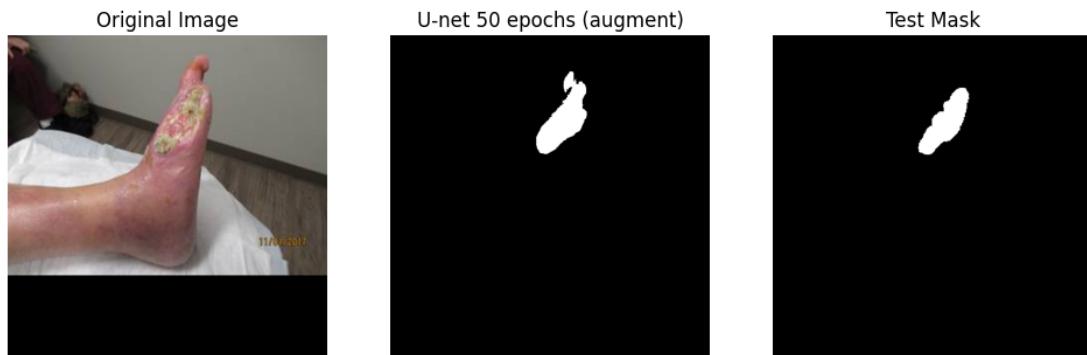
รูปที่ 4.41 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-net 50 epochs ที่ไม่มีการเพิ่มจำนวนกับ รูปทดสอบ



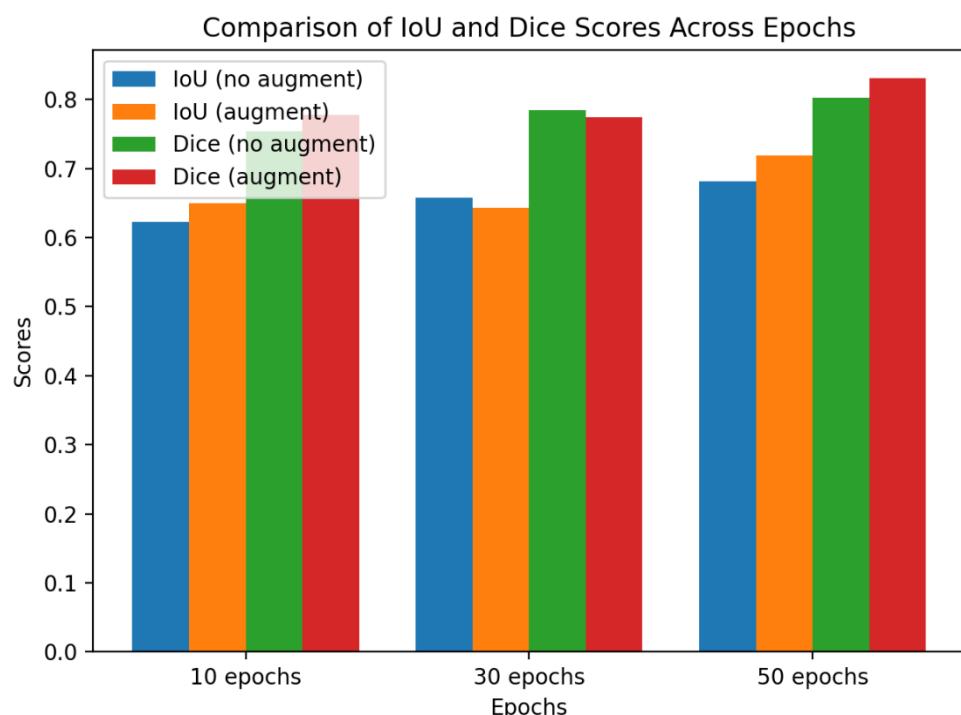
รูปที่ 4.42 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-net 10 epochs ที่มีการเพิ่มจำนวนกับ รูปทดสอบ



รูปที่ 4.43 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-net 30 epochs ที่มีการเพิ่มจำนวนกับ รูปทดสอบ



รูปที่ 4.44 รูปเปรียบเทียบหน้ากากที่ได้จากการทำนายของ U-net 50 epochs ที่มีการเพิ่มจำนวนกับรูปทดสอบ



รูปที่ 4.45 กราฟแท่งเปรียบค่าเมตริกของการทดลองที่ 2

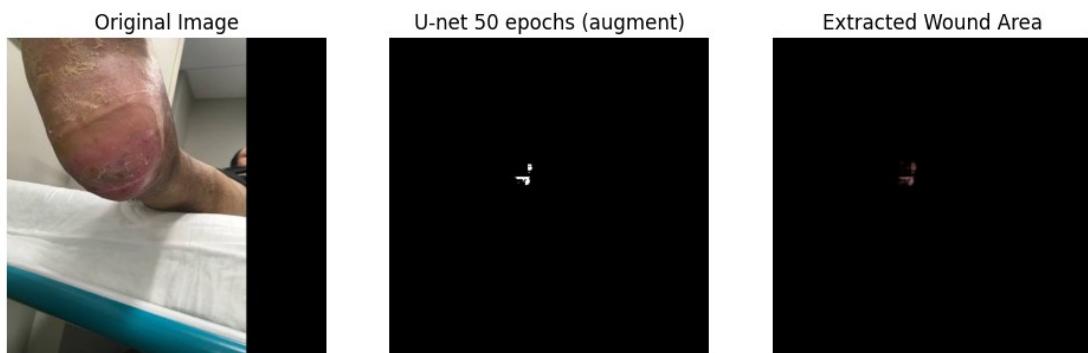
จากการทดลองรูปแบบที่ 2 เป็นการเปรียบเทียบประสิทธิภาพของโมเดล U-net ที่ฝึกสอนด้วยการเพิ่มข้อมูล (Data Augmentation) เทียบกับโมเดลที่ฝึกสอนโดยไม่เพิ่มข้อมูล โดยใช้ เมตริก IoU และ Dice Coefficient เพื่อวัดความแม่นยำในการทำนายผลลัพธ์ พบว่าการเพิ่มข้อมูล ช่วยปรับปรุงประสิทธิภาพของโมเดลในหลายรอบของการฝึก (epochs)

ผลการทดลองแสดงให้เห็นว่า โมเดลที่ใช้การเพิ่มข้อมูลมีค่า IoU และ Dice สูงกว่า โมเดลที่ไม่เพิ่มข้อมูลในหลายช่วงการฝึก โดยเฉพาะในรอบที่ 50 epochs ค่า IoU เพิ่มขึ้นจาก 0.6814 เป็น 0.7189 และค่า Dice เพิ่มขึ้นจาก 0.8030 เป็น 0.8303 และแสดงให้เห็นถึงความแม่นยำในการแบ่งส่วนภาพที่ดีขึ้นเมื่อใช้การเพิ่มข้อมูลอย่างไรก็ตามในบางรอบ เช่น ที่ 30 epochs พบร่วมค่า IoU ของ โมเดลที่ไม่เพิ่มข้อมูล (0.6571) กับสูงกว่าค่าที่เพิ่มข้อมูล (0.6434) ซึ่งอาจบ่งชี้ถึงการปรับจูนที่ยังไม่สมบูรณ์ในบางช่วงของการฝึก

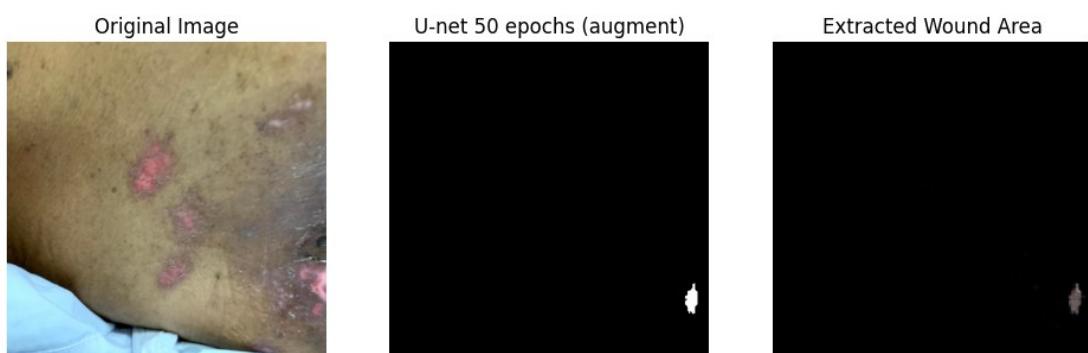
โดยสรุป การเพิ่มข้อมูล (Data Augmentation) ช่วยเพิ่มประสิทธิภาพของโมเดล U-net โดยเฉพาะอย่างยิ่งในการฝึกหลายรอบ (epochs) ที่มีการปรับจูนโมเดลต่อไป

#### 4.4.3.5 ปัญหาที่พบจากการทดลองแยกส่วนภาพ (segmentation)

จากการทดสอบ โมเดลแยกส่วนภาพ พบร่วมแพลงใน Class 1 และ Class 2 ยังคงให้ผลลัพธ์ที่ไม่น่าพึงพอใจเมื่อผ่านการประมวลผลด้วย โมเดล segmentation



รูปที่ 4.46 ผลลัพธ์การทำงานของ U-net 50 epochs (augment) กับรูปแพลงใน class ที่ 1

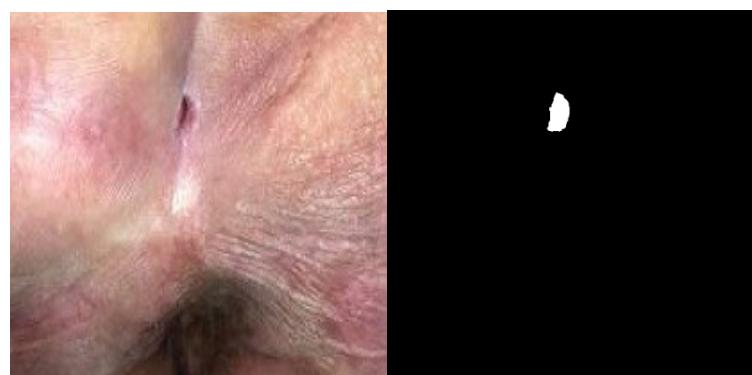


รูปที่ 4.47 ผลลัพธ์การทำงานของ U-net 50 epochs (augment) กับรูปแพลงใน class ที่ 2

สารเหตุของปัญหาดังกล่าวอาจเกิดจากความไม่สมดุลของชุดข้อมูลที่ใช้ฝึกสอน โมเดล segmentation ซึ่งนำมารากชุดข้อมูล "Wound Images Segmentation" บนแพลตฟอร์ม Kaggle โดยชุดข้อมูลดังกล่าวมีสัดส่วนของภาพแพลงใน Class 3 และ Class 4 มากกว่า Class 1 และ Class 2 ส่งผลให้โมเดลเรียนรู้ลักษณะของแพลงใน Class 3 และ Class 4 ได้ดี แต่ขาดความแม่นยำในการจำแนกแพลงใน Class 1 และ Class 2 ซึ่งเป็นประเภทที่มีจำนวนน้อยในชุดข้อมูลฝึกสอน

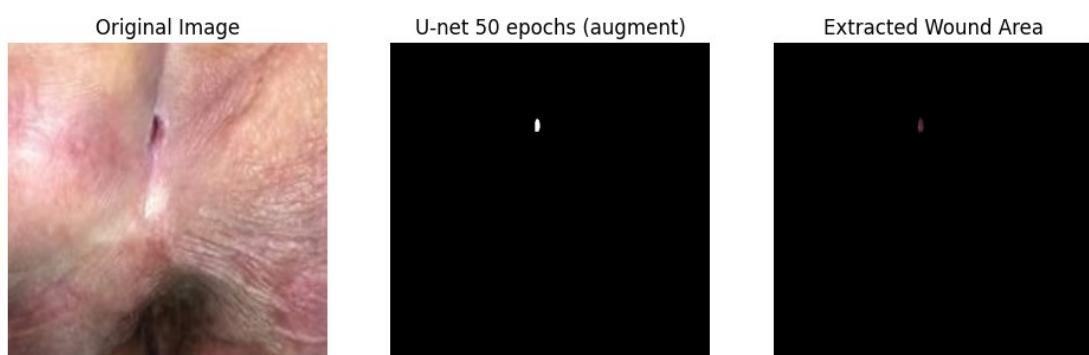
#### 4.4.3.6 การแก้ไขปัญหาในส่วนของ Segmentation Model

เพื่อแก้ไขปัญหาความไม่แม่นยำของโมเดล segmentation สำหรับแพลงใน Class 1 และ Class 2 ได้มีการเพิ่มชุดข้อมูลฝึกสอนโดยได้รับความร่วมมือจากผู้เชี่ยวชาญด้านแพทย์ด้านใน การสร้าง label และใช้เทคนิคในการเพิ่มข้อมูล (augmentation) สำหรับแพลงประเภทดังกล่าว ส่งผลให้จำนวนข้อมูลภาพที่ใช้ในการฝึกสอนเพิ่มขึ้นจาก 2,206 ภาพ เป็น 3,713 ภาพ

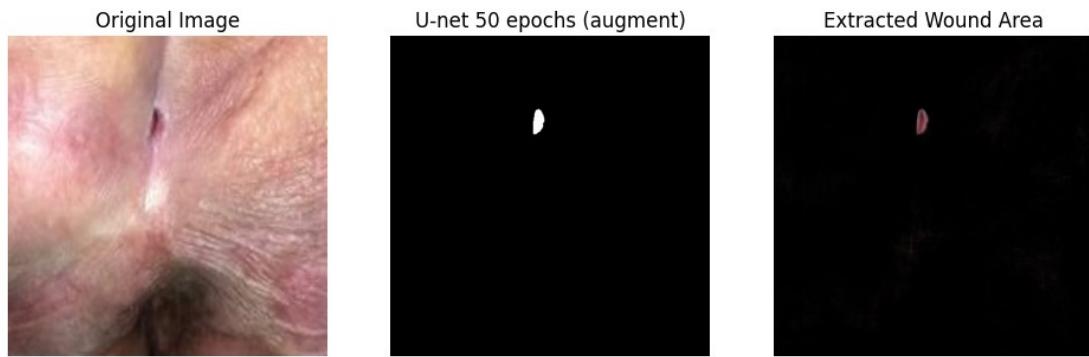


รูปที่ 4.48 ตัวอย่าง label ที่มีการจัดทำขึ้นโดยผู้เชี่ยวชาญ

หลังจากเพิ่มข้อมูลฝึกสอน ได้มีการทดลองเปรียบเทียบประสิทธิภาพของ segmentation model โดยพิจารณาผลลัพธ์ก่อนและหลังเพิ่มข้อมูล ทั้งในเชิงคุณภาพจากการที่ได้ หลังการแยกส่วน และเชิงปริมาณ โดยใช้ Intersection over Union (IoU) และ Dice Score เป็นตัวชี้วัดประสิทธิภาพของโมเดล



รูปที่ 4.49 ภาพของรูปแพลงใน class ที่ 1 และ 2 ก่อนเพิ่มข้อมูล



รูปที่ 4.50 ภาพของรูปแพลงใน class ที่ 1 และ 2 หลังเพิ่มข้อมูล

ตารางที่ 4.9 ตารางวัดผลค่าแมตริกเปรียบเทียบก่อนและหลังเพิ่มข้อมูลจากชุดข้อมูลแพลง Class ที่ 1 และที่ 2

Model	Class 1		Class 2	
	IoU	Dice	IoU	Dice
Unet 50 epochs ก่อนเพิ่มข้อมูล	0.432	0.601	0.518	0.649
Unet 50 epochs หลังเพิ่มข้อมูล	0.683	0.817	0.714	0.844

ตารางที่ 4.10 ตารางวัดผลค่าแมตริกเปรียบเทียบก่อนและหลังเพิ่มข้อมูลจากชุดข้อมูลแพลงทั้งหมด

Model	IoU	Dice
Unet 50 epochs ก่อนเพิ่มข้อมูล	0.7189	0.8303
Unet 50 epochs หลังเพิ่มข้อมูล	0.7014	0.817

#### 4.4.3.7 สรุปผล segmentation model

จากการทดลองแบ่งส่วนภาพ (Segmentation) สำหรับภาพแพลโดยใช้โมเดล U-Net และ MobileNetV2 พบว่า U-Net มีประสิทธิภาพสูงกว่า โดยให้ค่า IoU และ Dice สูงกว่า MobileNetV2 อย่างมีนัยสำคัญ จึงเหมาะสมกับงานนี้มากกว่า นอกจากนี้ การเพิ่มข้อมูล (Data Augmentation) เช่น การหมุนภาพ การกลับด้าน และการบีบบัดสี ช่วยให้โมเดลเรียนรู้ลักษณะแพลได้ดีขึ้น โดยค่า IoU และ Dice สูงขึ้น เมื่อจำนวนรอบการฝึกเพิ่มขึ้น

อย่างไรก็ตาม โมเดลยังมีปัญหาในการจำแนกแพลงประเกท (Class 1 และ 2) เนื่องจาก ความไม่สมดุลของชุดข้อมูล ซึ่งทำให้โมเดลเรียนรู้แพลงประเกทได้น้อยกว่าที่ควร แก้ปัญหานี้ทำโดย เพิ่มข้อมูลฝึกสอนใน Class 1 และ 2 ส่งผลให้ประสิทธิภาพของโมเดลดีขึ้นมากกว่า 25% ในบางกรณี

โดยสรุป U-Net เป็นโมเดลที่เหมาะสมที่สุด สำหรับงานนี้ การใช้ Data Augmentation ช่วยเพิ่มความแม่นยำ และการแก้ปัญหาความไม่สมดุลของข้อมูลทำให้โมเดลสามารถแยกแพลงประเกทด้วย ได้ดีขึ้น ซึ่งเป็นพื้นฐานสำคัญสำหรับการพัฒนาโมเดลจำแนกประเกทแพล (Classification Model) ในอนาคต

#### 4.4.4 กระบวนการพัฒนาโมเดลสำหรับจำแนกประเกทภาพ (Classification Model)

##### 4.4.4.1 การเปรียบเทียบประสิทธิภาพของโมเดลสำหรับจำแนกประเกทภาพ

###### (Classification Model)

ในการทดสอบนี้ เราได้ทำการเปรียบเทียบประสิทธิภาพของโมเดลสำหรับจำแนกประเกทภาพจำนวนสองโมเดล ได้แก่ ResNet และ EfficientNet ในการจำแนกประเกทภาพแพล โดยใช้เมทริกที่นิยมใช้ในการประเมินประสิทธิภาพการจำแนกประเกทภาพ ได้แก่ เมทริกซ์ F1 Score ซึ่งเป็นเมทริกซ์ที่ใช้วัดความแม่นยำของโมเดลโดยรวม โดยคำนวณจาก precision และ recall เป็นประโยชน์เมื่อมีปัญหารือว่า class imbalance

ทั้งสองโมเดลจะถูกเปรียบเทียบเพื่อระบุว่าโมเดลใดมีประสิทธิภาพที่ดีกว่าในการจำแนกประเกทภาพแพลเพื่อการใช้งานในอนาคต โดยในขั้นตอนการพัฒนาโมเดลสำหรับจำแนกประเกทภาพ (Classification Model) ได้แก่

การทดลองที่ 1: เปรียบเทียบประสิทธิภาพของโมเดลที่นำมาใช้ เพื่อวิเคราะห์ความแตกต่างในประสิทธิภาพของแต่ละโมเดล

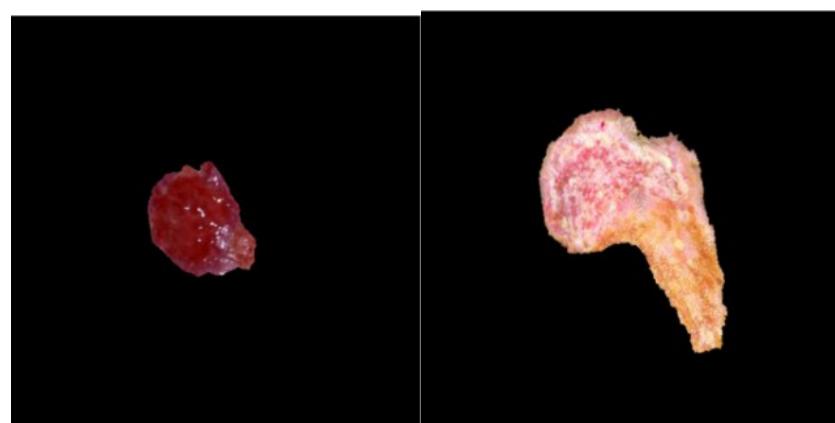
#### 4.4.4.2 สภาพแวดล้อมในการทดลอง

##### 4.4.4.2.1 ชุดข้อมูล (Dataset) สำหรับการทดลอง

การเก็บรวบรวมข้อมูลชุดนี้มาจากการแหล่งข้อมูลหลากหลายบนอินเทอร์เน็ต โดยได้รับความช่วยเหลือจากผู้เชี่ยวชาญในการแยกประเภทของแพลงก์ตอนและกรองภาพที่ไม่เกี่ยวข้องออกจากชุดข้อมูล โดยเมื่อกรองข้อมูลเสร็จสิ้น ชุดข้อมูลที่ได้ถูกแบ่งออกเป็นประเภทต่าง ๆ ดังนี้ การเก็บรวบรวมข้อมูลชุดนี้มาจากการแหล่งข้อมูลหลากหลายบนอินเทอร์เน็ต โดยได้รับความช่วยเหลือจากผู้เชี่ยวชาญในการแยกประเภทของแพลงก์ตอนและกรองภาพที่ไม่เกี่ยวข้องออกจากชุดข้อมูล โดยชุดข้อมูลจากอินเทอร์เน็ตนี้มีจำนวน 2206 รูปและชุดข้อมูลที่ได้จากผู้เชี่ยวชาญและผ่านการเพิ่มข้อมูล(augmentation)มีจำนวน 1010 รูป โดยเมื่อกรองข้อมูลเสร็จสิ้น ชุดข้อมูลที่ได้ถูกแบ่งออกเป็นประเภทต่าง ๆ ดังนี้

ตารางที่ 4.11 ตารางจำนวนข้อมูลแพลงในแต่ละระดับ

ประเภทแพลง	จำนวน
Class 1 (แพลงรุนแรงระดับ 1)	46
Class 2 (แพลงรุนแรงระดับ 2)	272
Class 3 (แพลงรุนแรงระดับ 3)	2085
Class 4 (แพลงรุนแรงระดับ 4)	613



รูปที่ 4.51 ตัวอย่างรูปแพลงที่นำไปฝึกสอนโมเดลสำหรับจำแนกประเภทภาพ

เนื่องจากชุดข้อมูลนั้นไม่สมดุล เราจึงต้องมีการปรับให้มันสมดุลเพื่อให้โมเดล classification นั้นเกิดการเรียนรู้อย่างถูกต้องไม่มีแนวโน้มทำนายไปใน class ใดคลาสหนึ่งมากกว่า

#### 4.4.4.2.2 การตั้งค่าพารามิเตอร์ในโมเดลสำหรับการทดลอง

ตารางที่ 4.12 ค่าพารามิเตอร์ที่ใช้ในการเทรนโมเดลในการทดลอง

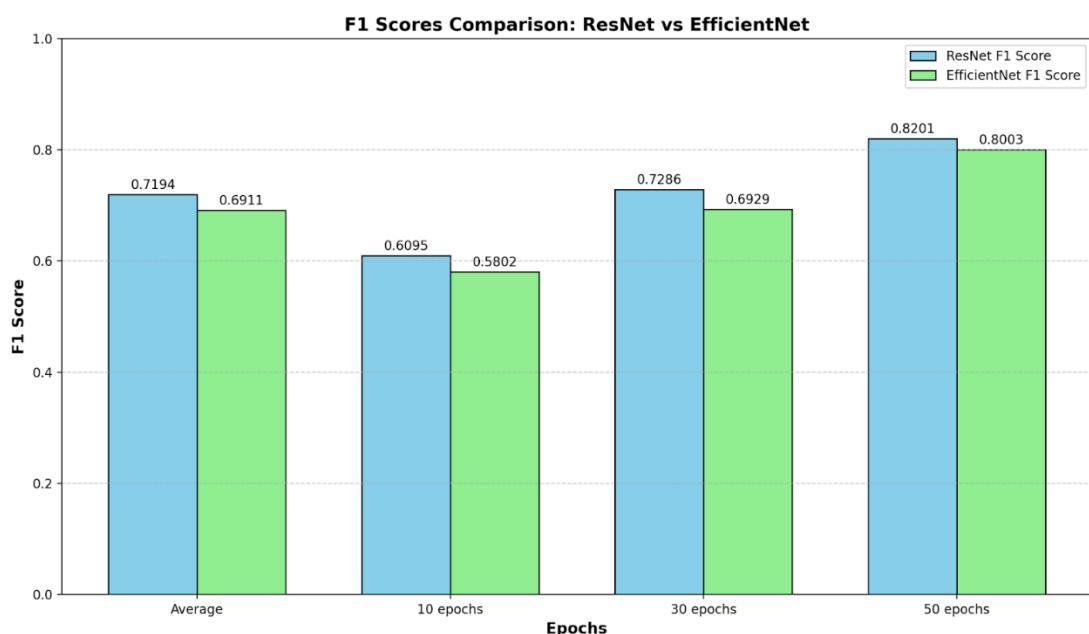
พารามิเตอร์	ค่าพารามิเตอร์
number_of_class	4 class
number_of_dataset (4 class)	2708 images
number_of_test_dataset (4 class)	308 images
classification_model	ResNet, EfficientNet
image_size	512x512
num_epochs	10,30,50
batch_size	32

#### 4.4.4.3 ผลการทดลองจำแนกประเภทภาระห่วงโมเดล ResNet และ EfficientNet

ในการทดลองนี้ได้ทดลองเปรียบเทียบผลในโมเดลสำหรับจำแนกภาพโดยใช้ชุดข้อมูลที่ได้มีการแบ่งคลาสโดยผู้เชี่ยวชาญและผ่านการแยกส่วน (segmentation) ด้วยโมเดล Unet 50 epochs จากนั้นทำการวัดผลด้วยค่าเมทริก F1 Score ซึ่งผลการทดลองเปรียบเทียบแสดงดังนี้

ตารางที่ 4.13 ตารางวัดผลค่าแมetrิกของการทดลองจำแนกประเภทภาระห่วงโน้มเดล ResNet และ EfficientNet

รอบ epochs	ResNet	EffcientNet
	F1 Score	F1 Score
ค่าเฉลี่ยรวม (Average)	0.7194	0.6911
10 epochs	0.6095	0.5802
30 epochs	0.7286	0.6929
50 epochs	0.8201	0.8003



รูปที่ 4.52 กราฟแท่งเปรียบค่าแมetrิกของการทดลองจำแนกประเภทภาระห่วงโน้มเดล ResNet และ EfficientNet

#### 4.4.4.4 สรุปผล segmentation model

จากการศึกษาประสิทธิภาพของโน้มเดลจำแนกประเภทภาระห่วง ResNet และ EfficientNet โดยใช้ชุดข้อมูลภาพแพลงก์ทับที่ผ่านการแบ่งประเภทโดยผู้เชี่ยวชาญและการแยกส่วนด้วยโน้มเดล Unet พบว่าโน้มเดล ResNet มีค่า F1 Score เนลีบสูงกว่าโน้มเดล EfficientNet ในทุกช่วงของการฝึกสอน (10, 30 และ 50 epochs) และคงให้เห็นถึงประสิทธิภาพที่เหนือกว่าในการจำแนกประเภทของแพลง กองจากนี้ ผลการทดลองยังแสดงให้เห็นว่าเมื่อจำนวน epochs เพิ่มขึ้น ค่า

F1 Score ของทั้งสองโมเดลนี่แนวโน้มเพิ่มขึ้นอย่างมีนัยสำคัญ อย่างไรก็ตาม โมเดล ResNet ยังคงมีค่าความแม่นยำสูงกว่า EfficientNet ในทุกช่วงของการฝึกสอน โดยเฉพาะที่ 50 epochs ซึ่ง ResNet มีค่า F1 Score เท่ากับ 0.8201 ขณะที่ EfficientNet มีค่า 0.8003

จากการทดลองดังกล่าว สามารถสรุปได้ว่า โมเดล ResNet มีประสิทธิภาพในการจำแนกประเภทภาพแพลก์ทับ ได้ดีกว่า EfficientNet ตามค่าเมตริก F1 Score ที่สูงกว่าในทุกรุ่น ดังนั้น โมเดล ResNet จึงเป็นตัวเลือกที่เหมาะสมกว่า สำหรับการนำไปใช้งานจริงในการจำแนกประเภทของแพลก์ทับ

## 4.5 การติดตั้งใช้งาน (Deployment) ระบบในภาพรวม

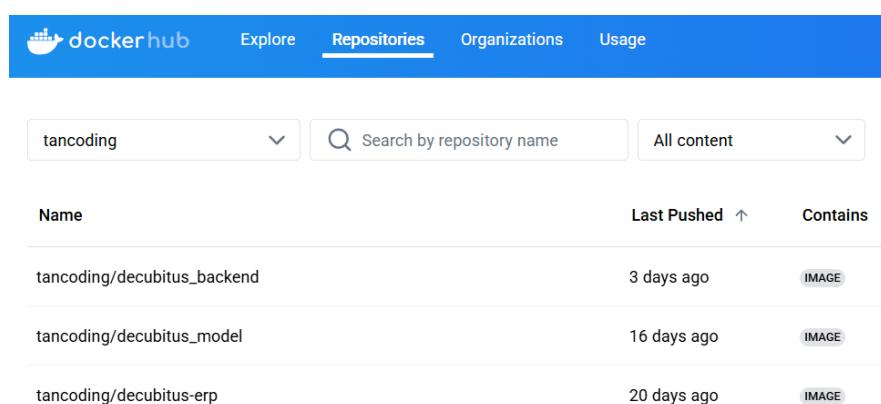
กระบวนการติดตั้งและใช้งานระบบ (Deployment) ได้รับการออกแบบให้รองรับ Containerization เพื่อเพิ่มความคล่องตัวในการพัฒนาและปรับใช้ระบบ โดยใช้เทคโนโลยี Docker Kubernetes และ Cloudflare Tunnel เพื่อให้ระบบสามารถทำงานได้อย่างเสถียร ยืดหยุ่น และรองรับการเข้าถึงจากภายนอกอย่างปลอดภัย

### 4.5.1 การปรับปรุงระบบให้สามารถทำงานในรูปแบบ Container

ระบบได้รับการออกแบบให้ทำงานภายใน Container เพื่อเพิ่มความสะดวกในการจัดการและขยายขนาด โดยใช้ NestJS ในส่วนของระบบหลังบ้าน (Backend) และ Angular ในส่วนของระบบหน้าบ้าน(Frontend)ในส่วนของระบบ ERP และ FastAPI สำหรับเป็น API ที่รองรับการประมวลผลของโมเดลและ PostgreSQL เป็นฐานข้อมูลหลักของระบบ

### 4.5.2 การจัดการ Containerization โดย Docker Hub

ระบบใช้ Docker Hub เป็น Container Registry สำหรับจัดเก็บและการจัดการ Docker Images เพื่อให้สามารถดึงและใช้งานได้อย่างเป็นระบบ การจัดการนี้ช่วยให้ควบคุมเวอร์ชันของซอฟต์แวร์ได้อย่างมีประสิทธิภาพ



รูปที่ 4.53 รูปภาพ Docker Images ใน Docker Hub

#### 4.5.3 การออกแบบไฟล์ตั้งค่าสำหรับ Kubernetes ใน Server Lab

ระบบใช้ Kubernetes บริหารจัดการ Container Orchestration โดยกำหนด Deployment และ Service เพื่อเพิ่มความเสถียรและรองรับการขยายตัว โดยกำหนด ReplicaSet ให้มี Pod ทำงานอยู่เสมอ หาก Pod ล่ม Kubernetes จะสร้างใหม่อัตโนมัติผ่าน Restart Policy (Always) พร้อมใช้ Liveness Probe เพื่อตรวจสอบสุขภาพของ Container และ Readiness Probe เพื่อกำหนดรับ Traffic เนื่องจาก Pod ที่พร้อมใช้งาน ช่วยลด Downtime และเพิ่มความเสถียรของระบบ

```

Last login: Sun Feb 23 14:52:38 2025 from 192.168.89.206
decubitus@orion: ~ + ~
decubitus@orion: $ kubectl get all -n decubitus
NAME                                         READY   STATUS    RESTARTS   AGE
pod/angular-frontend-bf79d6fc6-szvuk          1/1    Running   0          7d20h
pod/angular-frontend-bf79d6fc6-wskjt         1/1    Running   0          18d
pod/cloudflare-tunnel-6dcfd4594d-19cv8       1/1    Running   0          7d20h
pod/fastapi-gpu-deployment-564bd47655-kpwn4   1/1    Running   0          15d
pod/nestjs-backend-66595f99b-f4xwd            1/1    Running   0          2d23h
pod/postgres-db-cd5cb6b69-vnnvh              1/1    Running   0          11d

NAME                           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/backend-service        ClusterIP  10.152.183.198 <none>        80/TCP    36d
service/cloudflare-tunnel-service ClusterIP  10.152.183.216 <none>        80/TCP    28d
service/db                     ClusterIP  10.152.183.72  <none>        5432/TCP  41d
service/fastapi-service        ClusterIP  10.152.183.176 <none>        80/TCP    16d
service/frontend-service       ClusterIP  10.152.183.81  <none>        80/TCP    20d
service/postgres-db           ClusterIP  10.152.183.195 <none>        5432/TCP  41d

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/angular-frontend  2/2    2           2           20d
deployment.apps/cloudflare-tunnel 1/1    1           1           28d
deployment.apps/fastapi-gpu-deployment 1/1    1           1           16d
deployment.apps/nestjs-backend     1/1    1           1           41d
deployment.apps/postgres-db       1/1    1           1           41d

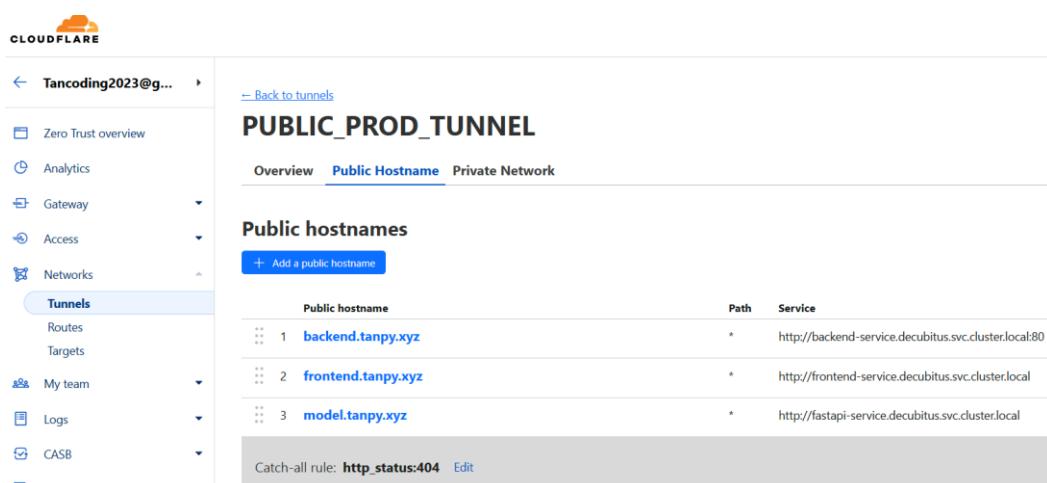
NAME                         DESIRED   CURRENT   READY   AGE
replicaset.apps/angular-frontend 2         2         2     20d
replicaset.apps/cloudflare-tunnel 1         1         1     20d
replicaset.apps/cloudflare-tunnel-6dcfd4594d 1         1         1     28d
replicaset.apps/cloudflare-tunnel-8c5774469b 0         0         0     28d
replicaset.apps/fastapi-gpu-deployment-564bd47655 1         1         1     16d

```

รูปที่ 4.54 ภาพการทำงานของ pod และ service ใน kubernetes server

#### 4.5.4 การตั้งค่า Cloudflare Tunnel เพื่อรับการเข้าถึงจากภายนอก

เพื่อให้ระบบเปิดให้บริการต่อสาธารณะอย่างปลอดภัย ได้มีการตั้งค่า Cloudflare Tunnel เพื่อเชื่อมต่อระบบภายในเซิร์ฟเวอร์กับ Public Internet โดยผ่าน network ของ Cloudflare กับ Public hostname ภายนอก



รูปที่ 4.55 รูปภาพ Public hostname ที่มีการตั้งค่าไว้ที่ Cloudflare

## บทที่ 5

# สรุปผลการดำเนินงาน

## 5.1 สรุปผลการดำเนินงาน

### 5.1.1 ระบบดูแลรักษาแพลก็อตทับ

พัฒนาระบบดูแลรักษาแพลก็อตทับประกอบด้วยระบบต่างๆ ดังนี้

1. รวบรวม Requirement
2. ออกแบบ UX/UI
3. ออกแบบ Use Case Diagram
4. ออกแบบ Sequence Diagram
5. ออกแบบ User Flow
6. ออกแบบ Database
7. ออกแบบสถาปัตยกรรมของระบบ
8. ระบบการสมัครสมาชิก
9. ระบบการเข้าสู่ระบบ
10. ระบบลืมรหัสผ่าน
11. ระบบการทำรายการตรวจแพลก็อต
12. ระบบสร้างรายการแพลก็อตของผู้ใช้งานทั่วไป
13. ระบบแทบทองผู้ใช้งานทั่วไป
14. ระบบการติดตามแพลก็อตของผู้ใช้งานทั่วไป
15. ระบบการเพิ่มผู้ป่วยในการดูแลของพยาบาล
16. ระบบการยืนยันแพลก็อต
17. ระบบแทบทองพยาบาล
18. ระบบในส่วนของผู้ดูแลระบบ
19. ระบบแสดงข้อมูลและจำนวนผู้ป่วยภายในระบบ

### 5.1.2 ระบบระบุและจำแนกความรุนแรงของแพลกตับ

พัฒนาระบบระบุและจำแนกความรุนแรงของแพลกตับ ได้ผลการศึกษาและพัฒนาต่างๆ ดังนี้

1. ศึกษางานวิจัยที่เกี่ยวข้องกับการทำ Classification model
2. ศึกษางานวิจัยที่เกี่ยวข้องกับการทำ Segmentation model
3. ศึกษาวิธีการทำ Classification model
4. ศึกษาวิธีการทำ Segmentation model
5. ระบบการรับรูปภาพจากมือถือเพื่อส่งไปยังตัวโมเดล
6. ทดลองเปรียบเทียบประสิทธิภาพของ โมเดลสำหรับจำแนกประเภทภาพ (Classification Model)
7. ทดลองเปรียบเทียบประสิทธิภาพของ โมเดลสำหรับการแบ่งส่วนภาพ (Segmentation Model)
8. ระบบส่งผลลัพธ์ไปเชื่อมต่อกับระบบคุณภาพแพลกตับ

### 5.2 ปัญหาและแนวทางแก้ไข

ปัญหาที่เกิดขึ้น

1. โปรเจกมีความซับซ้อนในการทำมาก ทำให้ต้องใช้เวลาในการศึกษาทำความเข้าใจ
2. Classification model ทำได้ยากจากเนื้องจากรูปภาพแพลกตับในระดับที่สูงมีความใกล้เคียงกันมาก

แนวทางการแก้ปัญหา

1. แบ่งเวลาในการศึกษาให้ชัดเจน เพื่อที่จะให้การศึกษาการทำโปรเจกต์ไม่ไปกระทบงานในส่วนอื่นๆ
2. หาส่วนประกอบอื่นมาช่วยในการจำแพลกตับในระดับที่สูง

### 5.3 แนวทางการพัฒนาต่อ

1. พัฒนาระบบ Segmentation model ให้มีความแม่นยำในการทำงานมากขึ้น
2. พัฒนาระบบ Classification model ให้สามารถทำงานได้ และมีประสิทธิภาพ
3. ทำการเชื่อมต่อข้อมูลภายนอกเพื่อปรับปรุงการทำงาน เช่น การนำข้อมูลในโรงพยาบาลเพื่อเป็นการติดตามข้อมูลของคนไข้ได้อย่างสะดวกและถูกต้องมากขึ้น

## បរចាំនូករម

Nico Curti, Yuri Merli, Corrado Zengarini ,Enrico Giampieri, Alessandra Merlotti ,Daniele Dall’Olio ,Emanuela Marcelli ,Tommaso Bianchi andGastone Castellani . 2023 .Effectiveness of Semi-Supervised Active Learning in Automated Wound Image Segmentation.[Online]. Available:<https://www.mdpi.com/1422-0067/24/1/706>

Chuanbo Wang, D. M. Anisuzzaman, Victor Williamson, Mrinal Kanti Dhar, Behrouz Rostami, Jeffrey Niezgoda, Sandeep Gopalakrishnan andZeyun Yu . 2020 .Fully automatic wound segmentation with deep convolutional neural networks.[Online].Available:  
<https://www.nature.com/articles/s41598-020-78799-w>

S. R. Oota et al, Wang, C et al and Thomas, S. et al . Wound Images Segmentation [2760 samples] .2024. [Online]. Available:  
<https://www.kaggle.com/datasets/leoscode/wound-segmentation-images>

ພວ.នគនី ផ្លើងសារិកិច.ការគ្រួលដោនគុណភាពរបៀប Pressure Ulcer Management.2019.  
[Online].Available:  
[https://www.crhospital.org/web\\_nurse/%E0%B9%81%E0%B8%9C%E0%B8%A5%E0%B8%81%E0%B8%94%E0%B8%97%E0%B8%B1%E0%B8%9A%E0%B8%AA%E0%B8%AD%E0%B8%99%202562%20RM.pdf](https://www.crhospital.org/web_nurse/%E0%B9%81%E0%B8%9C%E0%B8%A5%E0%B8%81%E0%B8%94%E0%B8%97%E0%B8%B1%E0%B8%9A%E0%B8%AA%E0%B8%AD%E0%B8%99%202562%20RM.pdf)

ការពៀងកាន់នៃរក្សាយាមផែកគុណភាពរបៀប: នាវប្លិបុត្រីអ៊ាងអិនបាបយ៉ា. 2019 .[Online]. Available:  
<https://thaietnurse.com/index.php/news/press-release/71-20210420>