



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบินของบริษัท
สามารถคอมเทค จำกัด ในสนามบินสุวรรณภูมิ
Network Maintenance Management System for SMART
Comtech Co.,Ltd. at Suvarnabhumi airport

นางสาวสุดารัตน์ พงศ์ศุกล
นางสาวสุนิสา พรานกลาง

ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบินของบริษัท
สามารถคอมเทค จำกัด ในสนามบินสุวรรณภูมิ
Network Maintenance Management System for SMART
Comtech Co.,Ltd. at Suvarnabhumi airport

นางสาวสุดารัตน์ พงศ์ศุกกล
นางสาวสุนิสา พรานกลาง

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา ระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบินของ
บริษัท สามารถคอมเทค จำกัด ในสนามบินสุวรรณภูมิ

ชื่อ-สกุล นักศึกษา นางสาวสุดารัตน์ พงศ์ศุกล

นางสาวสุนิสา พรานกลาง

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมโทรคมนาคม

ชื่อ-สกุล อาจารย์นิเทศ ผศ.ดร. พิเชฐ ม่วงนวล

ชื่อ-สกุล ผู้นิเทศงาน นายรัฐวุฒิ กิจเงา

ชื่อสถานประกอบการ บริษัท สามารถคอมเทค จำกัด

บทคัดย่อ

โครงการนี้ได้นำเสนอการออกแบบและพัฒนาแอปพลิเคชันสำหรับการทำงานในรูปแบบต่าง ๆ ในแผนกเน็ตเวิร์คของบริษัท สามารถคอมเทค จำกัด เป็นระบบแอปพลิเคชันที่สามารถสร้างตารางงานการบำรุงรักษาอุปกรณ์เครือข่ายที่ต้องทำโดยผู้ดูแลงาน ระบบได้มีการนำเทคโนโลยีของแผนที่มาใช้ในการนำทางเพื่อไปปฏิบัติงานตามสถานที่ที่ต้องปฏิบัติงานบริเวณสนามบินสุวรรณภูมิ มีแบบฟอร์มสำหรับการบำรุงรักษาอุปกรณ์เครือข่าย แบบฟอร์มปัญหาที่เกิดขึ้นกับอุปกรณ์เครือข่าย และแบบฟอร์มการแจ้งเปลี่ยนอุปกรณ์ ระบบมีการแสดงข้อมูลรายละเอียดของสถานที่ต่าง ๆ โดยระบุอาคาร ห้อง ชื่ออุปกรณ์ ไอพีแอดเดรส หมายเลขเครื่องของอุปกรณ์ มีการแสดงรายละเอียดของปัญหาที่เคยเกิดขึ้นแล้ว เป็นระบบที่สามารถใช้งานกับคลังพัสดุได้โดยมีระบบการนำเข้าอุปกรณ์ การนำออกอุปกรณ์ มีการแสดงข้อมูลอุปกรณ์ในคลังพัสดุคือ จำนวนของอุปกรณ์แต่ละประเภท โมเดล และหมายเลขเครื่องของอุปกรณ์ มีการแสดงข้อมูลรายละเอียดการนำเข้าและนำออกของอุปกรณ์ในคลังพัสดุ ระบบได้มีการนำเทคโนโลยีการตรวจสอบคิวอาร์โค้ดผ่านมือถือมาใช้ในการระบุตัวอุปกรณ์ เพื่อสะดวกต่อการแจ้งเปลี่ยนอุปกรณ์เครือข่าย ทั้งหมดนี้เพื่อเป็นการตอบโจทย์สำหรับพนักงานในแผนกเน็ตเวิร์คของบริษัท สามารถคอมเทค จำกัด ให้สามารถปฏิบัติงานและบริหารจัดการคลังพัสดุได้สะดวกสบายมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Co-Operative Title: Network Maintenance Management System for
SAMART Comtech Co.,Ltd. at Suvarnabhumi airport

Student Intern Name: Miss Sudarat Phongsukon
Miss Sunisa Phranklang

Faculty: Engineering **Department:** Telecommunication Engineering

Advisor Name: Asst.Prof.Dr. Phichet Moungnoul

Mentor Name: Nuttawut Kijjao

Company: SAMART Comtech Co.,Ltd

ABSTRACT

This project presents the planning and developing application for all kind of working in networking department of Samart Comtech company. This application system can build scheduled network maintenance which have to do by intendants. The system uses GPS in navigation to the places around Suvarnabhumi airport. There is network maintenance form, problems with network devices form, and devices change notification form. Moreover, this system shows the details of places by identifying the buildings, rooms, device names, IP addresses, serial numbers, and the detail of previous problems. It can use with warehouse by connecting with import-export system. Then, it will show the details of devices in warehouse which are the number of devices, model, and serial number, and the details of importation and exportation of devices in warehouse. In addition, this system uses QR code verification technology in identifying the devices which comfortable for network devices notification. All of these are for meeting the needs of employees in networking department of Samart Comtech company to work and manage warehouse more comfortable.

กิตติกรรมประกาศ

การจัดทำโครงการระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบินของบริษัท สามารถคอมเทค จำกัด ในสนามบินสุวรรณภูมินี้ เป็นโครงการที่ทำให้พัฒนาการทำงานของพนักงานในบริษัทให้สามารถทำงานสะดวกยิ่งขึ้น โดยโครงการสามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับการสนับสนุนและคำแนะนำจากหลาย ๆ ส่วน ตลอดระยะเวลาที่ได้เข้าไปฝึกสหกิจศึกษา ได้จัดทำโครงการดังกล่าว เรียนรู้ประสบการณ์การทำงานต่าง ๆ ในแผนก ตั้งแต่วันที่ 6 มิถุนายน 2561 จนถึง 23 พฤศจิกายน 2561

การจัดทำโครงการนี้ จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากขาดการสนับสนุนและคำปรึกษาจากหลาย ๆ ส่วน

ผศ.ดร. พิเชฐ ม่วงนวล อาจารย์ที่ปรึกษาภาควิชาวิศวกรรมโทรคมนาคม ที่ให้คำปรึกษาในการจัดทำโครงการ และแก้ไขเมื่อเกิดปัญหาต่าง ๆ ในการจัดทำโครงการ รวมทั้งให้คำแนะนำที่ทำให้โครงการมีประสิทธิภาพยิ่งขึ้น

นายวสุวัฒน์ พรเลิศอริยโชค ผู้จัดการฝ่ายระบบเครือข่าย นายณัฐวุฒิ กิจเจา ผู้นิเทศงาน และพี่ ๆ ในบริษัท สามารถคอมเทค จำกัด ทุกคนที่ให้คำปรึกษา ให้ความรู้และความเข้าใจทั้งเรื่อง การปฏิบัติงานในบริษัท และการจัดทำโครงการ รวมทั้งให้คำปรึกษาวิธีการแก้ไขปัญหาและพัฒนาให้ระบบดีขึ้น สามารถใช้งานได้อย่างมีประสิทธิภาพมากที่สุด

สุดท้ายนี้ขอขอบคุณเพื่อนนักศึกษาและครอบครัว ที่คอยช่วยเหลือในการจัดทำโครงการ ตลอดเวลาการทำงานวันแรกจนวันที่โครงการเสร็จ รวมถึงผู้มีพระคุณทุกท่านที่มีได้เอ่ยนามไว้ ณ ที่นี้ ที่มีส่วนสำหรับความสำเร็จ จึงขอขอบคุณไว้ ณ โอกาสนี้

สุดาร์ตน์ พงศ์ศุกล

สุนิสา พรานกลาง

ผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินโครงการ	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่ 2	
แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	5
2.1 React Native	5
2.1.1 ข้อดีของ React Native	5
2.1.2 ข้อเสียของ React Native	5
2.2 ภาษา JavaScript	6
2.3 โปรแกรม Visual Studio Code	6
2.4 โปรแกรม Genymotion	7
2.5 โปรแกรม Xcode	8
2.6 ระบบปฏิบัติการแอนดรอยด์	11
2.7 ระบบปฏิบัติการไอโอเอส	14

สารบัญ (ต่อ)

	หน้า
2.8 ระบบฐานข้อมูล	15
2.8.1 MySQL	15
2.8.2 โปรแกรม XAMPP	16
2.8.3 ภาษา PHP	17
2.9 MARVEL APP	17
2.10 บาร์โค้ด (BARCODE)	18
2.10.1 ข้อดีของบาร์โค้ด	18
2.10.2 ข้อเสียของบาร์โค้ด	19
2.10.3 ประเภทของบาร์โค้ด	19
2.11 GOOGLE MAPS API	24
บทที่ 3 วิธีดำเนินการโครงการ	30
3.1 การออกแบบ	30
3.1.1 การออกแบบการทำงานของระบบโดยรวม	30
3.1.2 การออกแบบและสร้างตารางฐานข้อมูล	31
3.1.3 การออกแบบส่วนต่อประสานกับผู้ใช้ (USER INTERFACE) ของแอปพลิเคชัน	39
3.1.4 การออกแบบเลขครุภัณฑ์คิวอาร์โค้ดของอุปกรณ์ภายในห้องจัดเก็บพัสดุ	65
3.2 เครื่องมือที่ใช้ในการทดลอง	65
3.2.1 โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์	65
3.2.2 โทรศัพท์เคลื่อนที่ระบบปฏิบัติการไอโอเอส	66
3.3 การจัดเก็บผลการทดลอง	66
บทที่ 4 ผลการดำเนินงาน	67
4.1 การทดสอบแอปพลิเคชัน	67
4.1.1 ทดสอบใช้งานแอปพลิเคชันของหน้าต่าง Login	67
4.1.2 ทดสอบใช้งานแอปพลิเคชันของหน้าต่าง Preventive Maintenance	69

สารบัญ (ต่อ)

	หน้า
4.1.3 ทดสอบใช้งานแอปพลิเคชันของหน้าต่าง Corrective Maintenance	85
4.1.4 ทดสอบใช้งานแอปพลิเคชันของหน้าต่างคลังพัสดุ (Store)	100
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	114
5.1 สรุปผลการดำเนินงาน	114
5.2 ประโยชน์ของโครงการ	115
5.3 ปัญหาและอุปสรรค	115
5.4 ข้อเสนอแนะและแนวทางการพัฒนาต่อ	116
บรรณานุกรม	117
ภาคผนวก ก โค้ดโปรแกรมคำสั่งสำหรับสร้างหน้าต่างแอปพลิเคชัน	119
ภาคผนวก ข โค้ดโปรแกรมคำสั่งสำหรับติดต่อกับฐานข้อมูล	164

สารบัญตาราง

ตารางที่	หน้า	
1.1	แผนการดำเนินโครงการ	3



สารบัญรูป

รูปที่	หน้า	
2.1	โปรแกรม Visual Studio Code	6
2.2	โปรแกรม Genymotion	7
2.3	ตัวอย่างการจำลองโทรศัพท์มือถือ Google Nexus 6 8.0 – API 26	8
2.4	โปรแกรม Xcode	8
2.5	ค้นหาโปรแกรม Xcode	9
2.6	กด GET และ Install	9
2.7	หน้าต่างโปรแกรม Xcode	10
2.8	ตัวอย่างการจำลองโทรศัพท์มือถือ iPhone	11
2.9	โครงสร้างของแอนดรอยด์	12
2.10	โปรแกรมจัดการฐานข้อมูล MySQL	16
2.11	โปรแกรม XAMPP	16
2.12	Marvel App ออกแบบหน้าต่างสำหรับแอปพลิเคชัน	17
2.13	ISBN-13	19
2.14	Codabar	20
2.15	Code128	20
2.16	Code39	20
2.17	Code93	21
2.18	EAN Code	21
2.19	GS1-128 (EAN-128)	21
2.20	Interleaved 2 Of 5	21
2.21	ISSN	22
2.22	MSI Plessey	22
2.23	POSTNET Barcode	22
2.24	UPC Barcode	22
2.25	QR Code	23

สารบัญรูป (ต่อ)

	หน้า
2.26 Maxi Code	23
2.27 PDF417	24
2.28 Data Matrix	24
2.29 กต GET STARTED	25
2.30 เลือก Maps	26
2.31 ตั้งชื่อ Project	26
2.32 กรอกข้อมูล	27
2.33 เปิดใช้ API และบริการ	27
2.34 เลือกการใช้งาน API	28
2.35 เปิดการใช้งาน API	28
2.36 API Key	29
3.1 บล็อกไดอะแกรมของระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายใน สนามบินของบริษัท สามารถคอมเทค จำกัด ในสนามบินสุวรรณภูมิ	30
3.2 แผนภาพการออกแบบฐานข้อมูล	32
3.3 การสร้างฐานข้อมูล	33
3.4 ตาราง “building”	33
3.5 ตาราง “device”	34
3.6 ตาราง “device_change”	34
3.7 ตาราง “device_type”	35
3.8 ตาราง “form_work”	35
3.9 ตาราง “location”	36
3.10 ตาราง “login”	36
3.11 ตาราง “problem”	37
3.12 ตาราง “room”	37
3.13 ตาราง “store”	38
3.14 ตาราง “user”	38

สารบัญรูป (ต่อ)

	หน้า
3.15 ตาราง “work”	38
3.16 Marvel App ออกแบบหน้าตาต่างสำหรับแอปพลิเคชัน	39
3.17 ออกแบบหน้าตาต่างสำหรับ Login	40
3.18 แผนผังแสดงการทำงานของระบบ Login	41
3.19 ออกแบบหน้าตาต่างหลักของแอปพลิเคชัน	42
3.20 หน้าตาต่างรายการงานที่ต้องทำ	43
3.21 ใส่ Password ยืนยันก่อนเพิ่มหรือลบงาน	43
3.22 หน้าตาต่าง Add Schedule	44
3.23 หน้าตาต่างการปิกหมุดสถานที่	45
3.24 เลือกเปิดไปที่ Google Maps ของ Preventive Maintenance	45
3.25 หน้าตาต่าง Google Maps ของ Preventive Maintenance	46
3.26 หน้าตาต่างข้อมูลเพิ่มเติมของ Preventive Maintenance	47
3.27 หน้าตาต่างแบบฟอร์มการบำรุงรักษาเครือข่าย	47
3.28 หน้าตาต่างการส่งข้อมูลสถานะของอุปกรณ์ไปยังอีเมล	48
3.29 แผนผังแสดงการทำงานของระบบ Preventive Maintenance	49
3.30 หน้าตาต่างหลักของ Corrective Maintenance	50
3.31 หน้าตาต่างการเตรียมตัวเข้าปฏิบัติงานเบื้องต้น	51
3.32 หน้าตาต่างการค้นหาเส้นทาง	51
3.33 หน้าตาต่างการปิกหมุดสถานที่	52
3.34 เลือกเปิดไปที่ Google Maps ของ Corrective Maintenance	53
3.35 หน้าตาต่าง Google Maps ของ Corrective Maintenance	53
3.36 หน้าตาต่างข้อมูลเพิ่มเติมของ Corrective Maintenance	54
3.37 หน้าตาต่างกรอกปัญหาที่พบ	55
3.38 หน้าตาต่างการแจ้งเปลี่ยนอุปกรณ์	55
3.39 หน้าตาต่างตัวอ่านคิวอาร์โค้ด	56
3.40 หน้าตาต่างปัญหาที่เกิดขึ้นแล้ว	57

สารบัญรูป (ต่อ)

	หน้า
3.41 หน้าต่างการส่งข้อมูลการปัญหาผ่านอีเมล	57
3.42 แผนผังแสดงการทำงานของระบบ Corrective Maintenance	58
3.43 หน้าต่างหลักของคลังพัสดุ	59
3.44 หน้าต่างตัวอ่านคิวอาร์โค้ดนำเข้าคลังพัสดุ	60
3.45 หน้าต่างกรอกข้อมูลอุปกรณ์นำเข้าของคลังพัสดุ	60
3.46 หน้าต่างตัวอ่านคิวอาร์โค้ดนำออกจากคลังพัสดุ	61
3.47 หน้าต่างกรอกข้อมูลอุปกรณ์นำออกจากคลังพัสดุ	61
3.48 หน้าต่างคลังพัสดุแยกตามประเภท	62
3.49 หน้าต่างรายละเอียดของแต่ละประเภท	62
3.50 หน้าต่างรายละเอียดการนำเข้าและนำออกจากคลังพัสดุ	63
3.51 หน้าต่างการส่งข้อมูลนำเข้า - ออกไปยังอีเมล	63
3.52 แผนผังแสดงการทำงานของระบบคลังพัสดุ	64
3.53 การออกแบบคิวอาร์โค้ดเลขครุภัณฑ์ของอุปกรณ์	65
3.54 โทรศัพท์เคลื่อนที่ยี่ห้อ OPPO	65
3.55 โทรศัพท์เคลื่อนที่ยี่ห้อ Apple	66
4.1 หน้าต่าง Login	67
4.2 ฐานข้อมูลตาราง “Login”	68
4.3 หน้าต่าง Login เมื่อกรอก Username และ Password	68
4.4 การแจ้งเตือนในหน้าต่าง Login เมื่อกรอก Username และ Password	69
4.5 หน้าต่าง Home ของแอปพลิเคชัน	69
4.6 หน้าต่างเพิ่ม ลบ และแสดงงานของ Preventive Maintenance	70
4.7 หน้าต่างกรอก Password เพื่อเพิ่มงานสำหรับผู้ดูแลระบบ	71
4.8 หน้าต่างเพิ่มงาน (Add Schedule)	71
4.9 หน้าต่างเพิ่มวันที่	72
4.10 หน้าต่างเพิ่มเวลา	72
4.11 หน้าต่างเพิ่มสถานที่	73

สารบัญรูป (ต่อ)

	หน้า
4.12 หน้าต่างแจ้งเตือนเมื่อเพิ่มงานสำเร็จ	73
4.13 ฐานข้อมูลตาราง “work” ที่ถูกบันทึกข้อมูลการเพิ่มงาน	74
4.14 หน้าต่างแสดงงานที่ต้องทำเมื่อเพิ่มงานสำเร็จ	74
4.15 หน้าต่างแจ้งเตือนเมื่อผู้ดูแลระบบกรอก Password ไม่ถูกต้อง	75
4.16 หน้าต่างปิดหมดสถานที่บนแผนที่	76
4.17 หน้าต่างให้กดไปเปิดที่ Google Maps	76
4.18 ทดสอบใช้งานแอปพลิเคชันเพื่อกดไปเปิดที่ Google Maps	77
4.19 หน้าต่างการใช้ Google Maps Application เพื่อนำทาง	77
4.20 ทดสอบการใช้ Google Maps Application เพื่อนำทาง	78
4.21 หน้าต่างแสดงข้อมูลเพิ่มของสถานที่นั้น ๆ	78
4.22 หน้าต่างแบบฟอร์มการบำรุงรักษาเครือข่าย	79
4.23 ทดสอบการใช้แอปพลิเคชันเพื่อกรอกแบบฟอร์มการบำรุงรักษาเครือข่าย และบันทึกข้อมูล	80
4.24 ฐานข้อมูลตาราง “from_work” ที่ถูกบันทึกข้อมูลการกรอกแบบฟอร์ม การบำรุงรักษาเครือข่าย	80
4.25 หน้าต่างแจ้งเตือนถามเมื่อกดปุ่ม “เสร็จงาน”	81
4.26 หน้าต่างแจ้งเตือนเมื่อปิดงานสำเร็จแล้ว	81
4.27 หน้าต่างแสดงงานที่ต้องทำเมื่อปิดงานสำเร็จ	82
4.28 หน้าต่างกรอกอีเมลและเลือกเดือน	82
4.29 หน้าต่างแจ้งเตือนเมื่อส่งอีเมลสำเร็จ	83
4.30 หน้าต่างแสดงข้อความรายงาน PM ในอีเมล	83
4.31 หน้าต่างโปรแกรม Excel แสดงสถานะของงาน PM	84
4.32 หน้าต่างโปรแกรม Excel แสดงรายงานผลจากแบบฟอร์มการบำรุงรักษา เครือข่าย	84
4.33 หน้าต่าง Home ของแอปพลิเคชัน	85
4.34 หน้าต่างหลักของ Corrective Maintenance	86

สารบัญรูป (ต่อ)

	หน้า
4.35 หน้าต่างการเตรียมเพื่อเข้าปฏิบัติงานเบื้องต้น	87
4.36 หน้าต่างค้นหาเส้นทางเพื่อเข้าปฏิบัติงาน Corrective Maintenance	88
4.37 หน้าต่างปิกหมุดสถานที่บนแผนที่	88
4.38 หน้าต่างข้อมูลเพิ่มเติมของสถานที่นั้น ๆ	89
4.39 หน้าต่างกรอกปัญหาที่พบ	90
4.40 หน้าต่างแจ้งเตือนเมื่อกรอกปัญหาที่พบสำเร็จ	90
4.41 ฐานข้อมูลตาราง “problem” ที่ถูกบันทึกข้อมูลปัญหาที่พบ	91
4.42 หน้าต่างหลักของ Corrective Maintenance	91
4.43 หน้าต่างแสดงปัญหาที่เกิดขึ้นแล้ว	92
4.44 หน้าต่างกรอกอีเมลและเลือกเดือน	92
4.45 หน้าต่างแจ้งเตือนเมื่อส่งอีเมลสำเร็จ	93
4.46 หน้าต่างแสดงข้อความรายงานปัญหาที่เกิดขึ้นแล้วของงาน CM	93
4.47 หน้าต่างโปรแกรม Excel แสดงรายงานปัญหาที่เกิดขึ้นแล้วของงาน CM	94
4.48 หน้าต่างหลักของ Corrective Maintenance	95
4.49 หน้าต่างการแจ้งเปลี่ยนอุปกรณ์	95
4.50 หน้าต่างสแกนคิวอาร์โค้ดอุปกรณ์เดิม	96
4.51 หน้าต่างสแกนคิวอาร์โค้ดอุปกรณ์ใหม่	96
4.52 หน้าต่างการแจ้งเปลี่ยนอุปกรณ์เมื่อสแกน QR Code	97
4.53 หน้าต่างการแจ้งเตือนเมื่อกดบันทึกการเปลี่ยนอุปกรณ์	98
4.54 หน้าต่างข้อมูลเพิ่มเติมเปรียบเทียบเมื่อเปลี่ยนอุปกรณ์	99
4.55 ฐานข้อมูลตาราง “device” แสดงข้อมูลอุปกรณ์เดิม	99
4.56 ฐานข้อมูลตาราง “device” แสดงข้อมูลอุปกรณ์ใหม่	100
4.57 หน้าต่าง Home ของแอปพลิเคชัน	100
4.58 หน้าต่างหลักของ Store	101
4.59 หน้าต่างอ่าน QR Code การนำเข้าอุปกรณ์	102

สารบัญรูป (ต่อ)

	หน้า
4.60 แสดงการสแกน QR Code เพื่อนำเข้าอุปกรณ์	102
4.61 หน้าต่างการนำเข้าอุปกรณ์แสดงข้อมูลการสแกน QR Code และกรอกข้อมูลเพื่อนำเข้า	103
4.62 หน้าต่างแจ้งเตือนเมื่อกด “บันทึก” ข้อมูลนำเข้าอุปกรณ์	104
4.63 ฐานข้อมูลตาราง “store”	104
4.64 หน้าต่างหลักของ store	105
4.65 หน้าต่าง “คลังพัสดุ”	105
4.66 หน้าต่างรายละเอียดอุปกรณ์ในคลังพัสดุ	106
4.67 หน้าต่างอ่าน QR Code การนำออกอุปกรณ์	107
4.68 แสดงการสแกน QR Code เพื่อนำออกอุปกรณ์	107
4.69 หน้าต่างการนำออกอุปกรณ์แสดงข้อมูลการสแกน QR Code และกรอกข้อมูลเพื่อนำออกอุปกรณ์	108
4.70 หน้าต่างแจ้งเตือนเมื่อกด “บันทึก” ข้อมูลนำออกอุปกรณ์	109
4.71 หน้าต่าง “คลังพัสดุ”	109
4.72 หน้าต่าง “รายละเอียดการนำเข้า – ออก”	110
4.73 หน้าต่างกรอกอีเมลและเลือกเดือน	111
4.74 หน้าต่างแจ้งเตือนเมื่อส่งอีเมลสำเร็จ	111
4.75 หน้าต่างแสดงข้อความรายการคลังพัสดุ	112
4.76 หน้าต่างโปรแกรม Excel แสดงรายงานคลังพัสดุ (จำนวน)	112
4.77 หน้าต่างโปรแกรม Excel แสดงรายละเอียดการนำเข้าและนำออกของอุปกรณ์ในคลังพัสดุ	113

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบันการใช้เทคโนโลยีเพื่ออำนวยความสะดวกมีมากมาย ซึ่งส่วนใหญ่เป็นแอปพลิเคชันที่อยู่บนโทรศัพท์มือถือที่สามารถอำนวยความสะดวกให้กับทุกคน โดยสามารถจัดการสิ่งที่ต้องการได้ทุกที่ ทุกเวลา เช่น การทำธุรกรรมทางการเงิน การติดต่อสื่อสารโดยใช้อินเทอร์เน็ต การจองที่นั่งสำหรับชมภาพยนตร์ การสั่งซื้อสินค้า เป็นต้น ผู้จัดทำและผู้จัดการฝ่ายระบบเครือข่าย บริษัทสามารถคอมเทค จำกัด จึงได้มีการศึกษาและสร้าง “ระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบิน” โดยสามารถใช้งานผ่านแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์และผู้จัดการฝ่ายระบบเครือข่ายได้ขอให้จัดทำแอปพลิเคชันของระบบปฏิบัติการไอโอเอส เพื่ออนาคตอาจจะมีการใช้งานและพัฒนาในระบบปฏิบัติการดังกล่าวต่อไป

จากที่กล่าวมาข้างต้นนั้น เป็นที่มาของการออกแบบและสร้างโครงการ “ระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบิน” เพื่อตอบโจทย์ระบบการทำงานให้กับพนักงานได้มีความสะดวกสบายยิ่งขึ้น

1.2 วัตถุประสงค์

- 1) ศึกษาและพัฒนาฐานข้อมูลตำแหน่งของอุปกรณ์เครือข่ายภายในสนามบิน
- 2) ศึกษาและพัฒนาฐานข้อมูลระบบคิวอาร์โค้ดสำหรับการจัดการอุปกรณ์เครือข่าย
- 3) ศึกษาและพัฒนาแอปพลิเคชันการนำทางไปยังอุปกรณ์เครือข่ายภายในพื้นที่สนามบิน
- 4) ศึกษาและพัฒนากระบวนการจัดการในการซ่อมบำรุงเครือข่ายและคลังวัสดุอุปกรณ์
- 5) ศึกษา เรียนรู้ และฝึกฝนทักษะการประยุกต์ใช้งานภาษา JavaScript, PHP, และ SQL ในการออกแบบส่วนประสานกราฟิกกับผู้ใช้ และการจัดการฐานข้อมูลของระบบ

1.3 ขอบเขตของโครงการ

ทำการออกแบบและพัฒนาแอปพลิเคชันเพื่ออำนวยความสะดวกให้กับพนักงานในบริษัท ประกอบไปด้วย 3 ระบบในการใช้งาน ได้แก่

1) ระบบการทำงานการบำรุงรักษาเชิงป้องกัน (Preventive Maintenance) เป็นระบบที่สนับสนุนการปฏิบัติงานด้านบำรุงรักษาอุปกรณ์เครือข่ายให้กับพนักงาน ด้วยการนำเทคโนโลยีของ Google Map มาใช้ในการนำทางไปยังตำแหน่งของอุปกรณ์เครือข่ายภายในสนามบินสุวรรณภูมิ ระบบมีฟังก์ชันสำหรับการวางแผนงานให้ผู้ดูแลงาน มีฟังก์ชันสำหรับพนักงานเพื่อเข้าไปปฏิบัติงาน มีแบบฟอร์มสำหรับกรอกข้อมูลการบำรุงรักษาเครือข่ายเพื่อเก็บผลการปฏิบัติงาน และมีฟังก์ชันสำหรับการนำออกข้อมูลผลการบำรุงรักษาเครือข่ายโดยการส่งอีเมล ออกเป็นไฟล์ Excel

2) ระบบการทำงานการบำรุงรักษาเชิงแก้ไข (Corrective Maintenance) เป็นระบบที่สนับสนุนการทำงานปฏิบัติงานด้านการแก้ไขปัญหาที่เกิดขึ้นกับอุปกรณ์เครือข่ายให้กับพนักงาน ด้วยการนำเทคโนโลยีของ Google Map มาใช้ในการนำทางไปยังตำแหน่งของอุปกรณ์เครือข่ายที่เกิดปัญหา โดยใช้ฟังก์ชันค้นหาสถานที่ตำแหน่งของอุปกรณ์เครือข่ายภายในสนามบินสุวรรณภูมิ มีแบบฟอร์มการกรอกปัญหาที่เกิดขึ้นกับอุปกรณ์เพื่อเก็บผลการปฏิบัติงานและสามารถดูปัญหาที่เคยเกิดขึ้นแล้วจากการกรอกของพนักงานที่ปฏิบัติงาน มีฟังก์ชันการแจ้งเปลี่ยนอุปกรณ์ ด้วยการนำเทคโนโลยีการตรวจสอบคิวอาร์โค้ดมาใช้เพื่อระบุตัวอุปกรณ์และรายละเอียดของอุปกรณ์

3) ระบบคลังพัสดุ (Store) เป็นระบบที่ไว้สำหรับการนำเข้าและนำออกของอุปกรณ์หรือพัสดุในแผนก โดยการนำพัสดุเข้าหรือนำออก สามารถใช้เทคโนโลยีของคิวอาร์โค้ดเพื่ออ่านเลขครุภัณฑ์ในการระบุพัสดุ และกรอกข้อมูลเพิ่มเติมลงไปเพื่อเก็บรายละเอียดของพัสดุ ซึ่งสามารถดูจำนวนและรายละเอียดของพัสดุในท้องจัดเก็บพัสดุได้ ได้แก่ จำนวนของแต่ละประเภท, Model และ Serial Number ระบบสามารถดูรายละเอียดการนำเข้าและนำออกของพัสดุ โดยมีรายละเอียดได้แก่ สถานะการนำเข้าหรือนำออก, Serial Number, ประเภท, ผู้นำเข้าหรือนำออก, สถานที่นำไปใช้ และหมายเหตุ

1.4 วิธีการดำเนินโครงการ

ตารางที่ 1.1 แผนการดำเนินโครงการ

ลำดับ	หัวข้องาน	เดือนที่ 1		เดือนที่ 2		เดือนที่ 3		เดือนที่ 4	
1.1	ศึกษาภาษาและโปรแกรมที่ใช้เขียนแอปพลิเคชัน	√	√						
1.2	ศึกษาหลักการการทำงานของบาร์โค้ดและศึกษาการใช้งานของฐานข้อมูล	√	√						
1.3	ออกแบบหน้าตาสำหรับแอปพลิเคชัน	√	√						
1.4	สร้างหน้าต่างแอปพลิเคชันสำหรับการนำทางและกระบวนการบำรุงรักษาเชิงป้องกัน (Preventive Maintenance)		√	√					
1.5	สร้างหน้าต่างแอปพลิเคชันสำหรับการนำทางและกระบวนการบำรุงรักษาเชิงแก้ไข (Corrective Maintenance)			√	√				
1.6	สร้างระบบการนำทาง		√	√	√				
2.1	ศึกษาการใช้งานโปรแกรม Xcode				√				
2.2	สร้างหน้าต่างแอปพลิเคชันสำหรับงาน Preventive Maintenance และ Corrective Maintenance (แก้ไข)				√	√			
2.3	สร้างหน้าต่างแอปพลิเคชันสำหรับคลังวัสดุ อุปกรณ์ (Store)					√	√		
2.4	สร้างหน้าต่างแอปพลิเคชันสำหรับอ่านคิวอาร์โค้ด					√	√		
2.5	ออกแบบและสร้างเลขครุภัณฑ์สำหรับคลังวัสดุ อุปกรณ์					√	√		
2.6	ออกแบบและสร้างฐานข้อมูล					√	√		
2.7	นำหน้าต่างแอปพลิเคชันทั้งหมดเชื่อมกับฐานข้อมูล					√	√		
3.1	สร้างฟังก์ชันสำหรับการนำออกข้อมูลเพื่อรายงานผล						√	√	
3.2	ทดสอบระบบทั้งหมด						√	√	√
3.3	จัดเก็บผลการทดลอง						√	√	√
4	จัดทำเล่มรายงาน								√
									√

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ระบบที่มีฐานข้อมูลตำแหน่งของอุปกรณ์เครือข่ายภายในสนามบินและสามารถใช้งานได้จริง
- 2) ระบบที่มีฐานข้อมูลระบบคิวอาร์โค้ดสำหรับการจัดการอุปกรณ์เครือข่ายที่สามารถระบุรายละเอียดของตัวอุปกรณ์และพัสดุต่าง ๆ
- 3) แอปพลิเคชันที่สามารถนำทางไปยังอุปกรณ์เครือข่ายภายในพื้นที่สนามบินได้
- 4) แอปพลิเคชันที่สามารถบริหารจัดการในการซ่อมบำรุงเครือข่ายและคลังวัสดุอุปกรณ์ให้สะดวกสบาย และมีประสิทธิภาพ
- 5) ได้รับความรู้และได้ทักษะการประยุกต์ใช้งานภาษา JavaScript, PHP, และ SQL ในการออกแบบส่วนประสานกราฟิกกับผู้ใช้ และการจัดการฐานข้อมูลของระบบ



บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 React Native

React Native เป็น Open Source Framework สำหรับใช้พัฒนา Cross platform mobile apps ที่สร้างขึ้นโดย Facebook เพื่อให้ผู้พัฒนาแอปพลิเคชันบนโทรศัพท์มือถือสามารถพัฒนาแอปพลิเคชันทั้งใน Android และ iOS โดยใช้ JavaScript เป็นหลักในการพัฒนา

2.1.1 ข้อดีของ React Native

- 1) สามารถออกแบบ User Interface ได้ง่ายมาก
- 2) การตกแต่ง User Interface ใช้คำสั่งที่คล้ายกับการตกแต่งของ CSS
- 3) มี Module รองรับการใช้งานที่หลากหลายรูปแบบ
- 4) ประสิทธิภาพการทำงานเทียบเท่ากับการเขียนแบบ Native
- 5) การเพิ่มโมดูลไปยัง Project ของ Android และ iOS ใช้เพียงคำสั่ง react-native link ก็จะมีเพิ่มเข้าไปใน Project อัตโนมัติ
- 6) สามารถพัฒนาแอปพลิเคชันได้อย่างรวดเร็ว
- 7) สามารถใช้พัฒนาแอปพลิเคชันได้ทั้งระบบปฏิบัติการ Android และระบบปฏิบัติการ iOS

2.1.2 ข้อเสียของ React Native

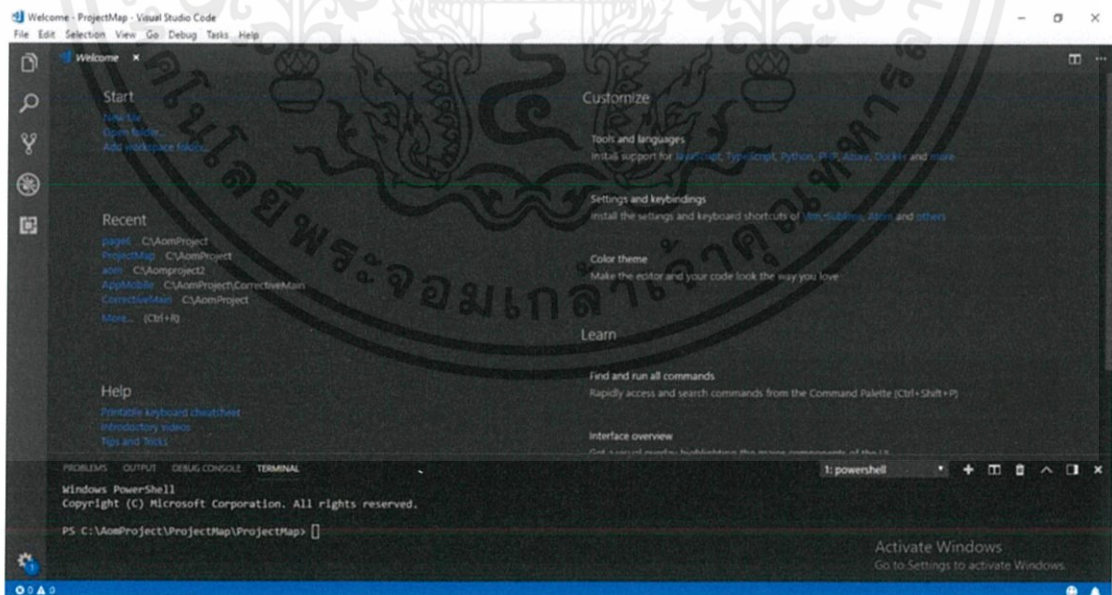
- 1) การเพิ่มโมดูลเข้าไปมาก ๆ จะทำให้การ Build ส่วนของ Android ช้ามาก เนื่องจากต้องไปคอมไพล์ในส่วนของโมดูลด้วย
- 2) โมดูลหลาย ๆ ตัวยังใช้ Build Tools Version เก่า ๆ ทำให้ต้องแก้ไข
- 3) บางครั้งตอนติดตั้งโมดูลใหม่เข้าไป จะทำการรีเซ็ตโมดูลเดิม ทำให้โค้ดที่เคยแก้ไขกลับไปเป็นเหมือนเดิมและต้องแก้ไข
- 4) การ Debug บางครั้งจะดูยาก [1]

2.2 ภาษา JavaScript

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ซึ่งในการสร้างและพัฒนาเว็บไซต์จะใช้ร่วมกับ HTML เพื่อให้เว็บไซต์ดูมีการเคลื่อนไหว และสามารถตอบสนองผู้ใช้งานได้มากขึ้น ส่วนมากจะใช้ JavaScript เขียนเป็นฟังก์ชันสำหรับการใช้งานต่าง ๆ เช่น ตรวจสอบความถูกต้องของข้อมูลในแบบฟอร์ม ใช้สร้างตัวเลือกเวลาและใช้สร้างตัวเลือกวันที่ในรูปแบบปฏิทิน เป็นต้น และยังสามารถใช้งานภาษา JavaScript ร่วมกับภาษา PHP ได้เป็นอย่างดี [2]

2.3 โปรแกรม Visual Studio Code

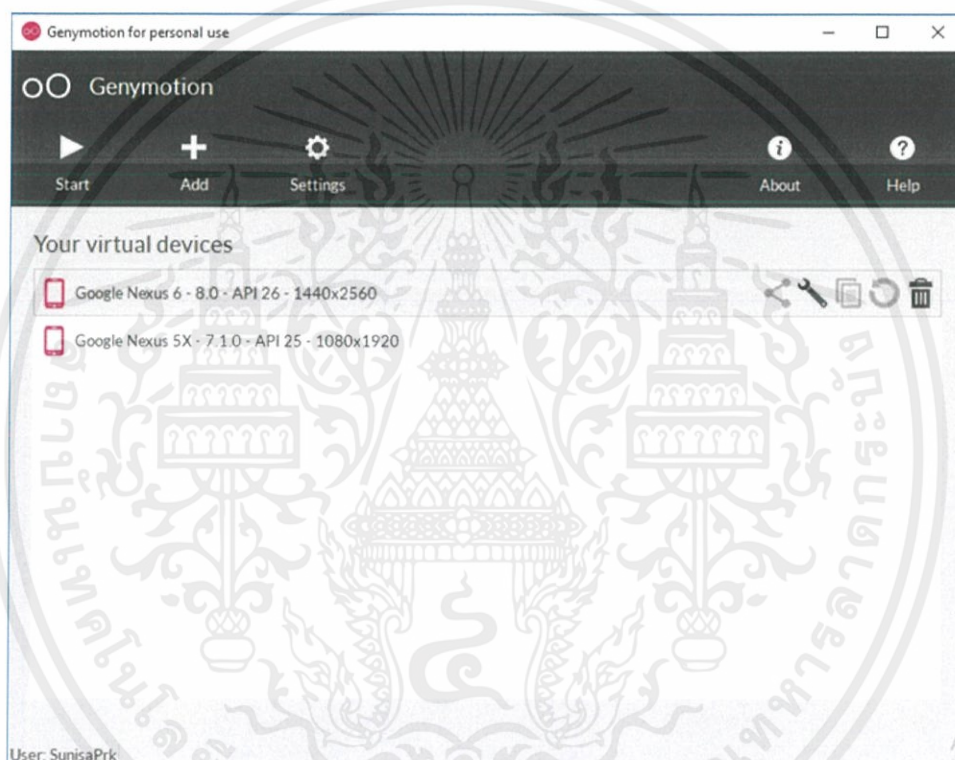
Visual Studio Code หรือ VS Code แสดงดังรูปที่ 2.1 เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ดจากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ Open Source ซึ่ง Visual Studio Code เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานแบบ Cross platform รองรับการใช้งานทั้งบน Windows, MacOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้งานอย่างมากไม่ว่าจะเป็นการเปิดใช้งานภาษาอื่น ๆ เช่น ภาษา C++, C#, Java, Python, PHP, Themes Debugger และ Commands เป็นต้น [3]



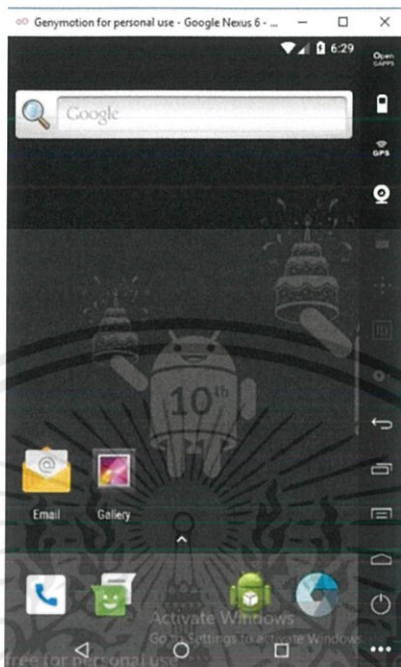
รูปที่ 2.1 โปรแกรม Visual Studio Code

2.4 โปรแกรม Genymotion

Genymotion แสดงหน้าต่างโปรแกรมดังรูปที่ 2.2 และรูปที่ 2.3 เป็น Android Emulator ที่มาพร้อมกับ Android System Image และ AVD สำเร็จรูป เพื่อช่วยให้นักพัฒนาไม่ต้องกังวลเรื่องการตั้งค่า AVD (Android Virtual Device) ที่สำคัญ ซึ่ง Genymotion ทำงานเร็วกว่า Emulator ที่มาพร้อมกับ Android SDK และ Genymotion มีเครื่องมือช่วยนักพัฒนาหลายอย่าง ทั้งการเชื่อมต่อกับ Webcam, GPS, Screencast เป็นต้น [4]



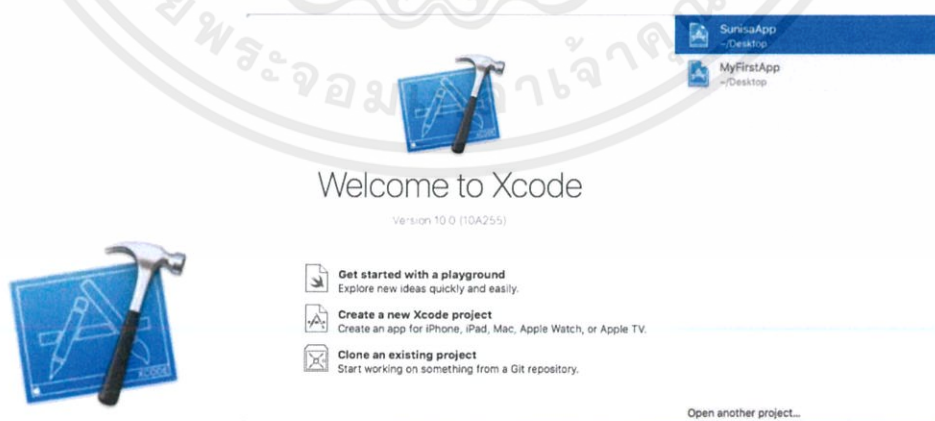
รูปที่ 2.2 โปรแกรม Genymotion



รูปที่ 2.3 ตัวอย่างการจำลองโทรศัพท์มือถือ Google Nexus 6 8.0 – API 26 บนโปรแกรม Genymotion

2.5 โปรแกรม Xcode

ในโครงการนี้ได้เลือกใช้โปรแกรม Xcode ดังรูปที่ 2.4 ให้เป็น Simulator ในการสร้าง iOS Application ซึ่งโปรแกรม Xcode ได้ถูกพัฒนาจากทาง Apple โดยออกแบบให้ใช้งานได้สะดวก รวดเร็วและยังสามารถเขียนโปรแกรมได้ [5]



รูปที่ 2.4 โปรแกรม Xcode

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการรึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

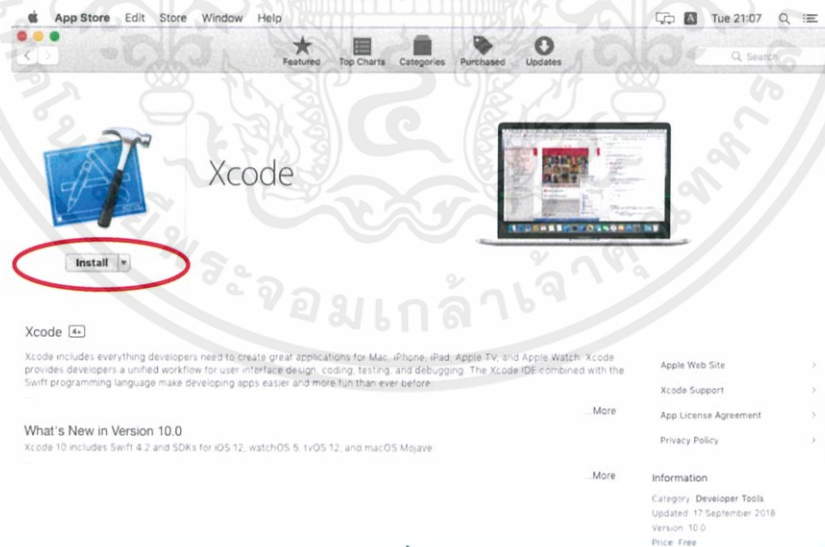
โปรแกรม Xcode สามารถดาวน์โหลดและติดตั้งบน Mac OS ซึ่งมีวิธีการติดตั้งดังนี้

1) ไปที่ App Store และค้นหา “Xcode” กดที่ GET และ Install ดังรูปที่ 2.5

และรูปที่ 2.6



รูปที่ 2.5 ค้นหาโปรแกรม Xcode

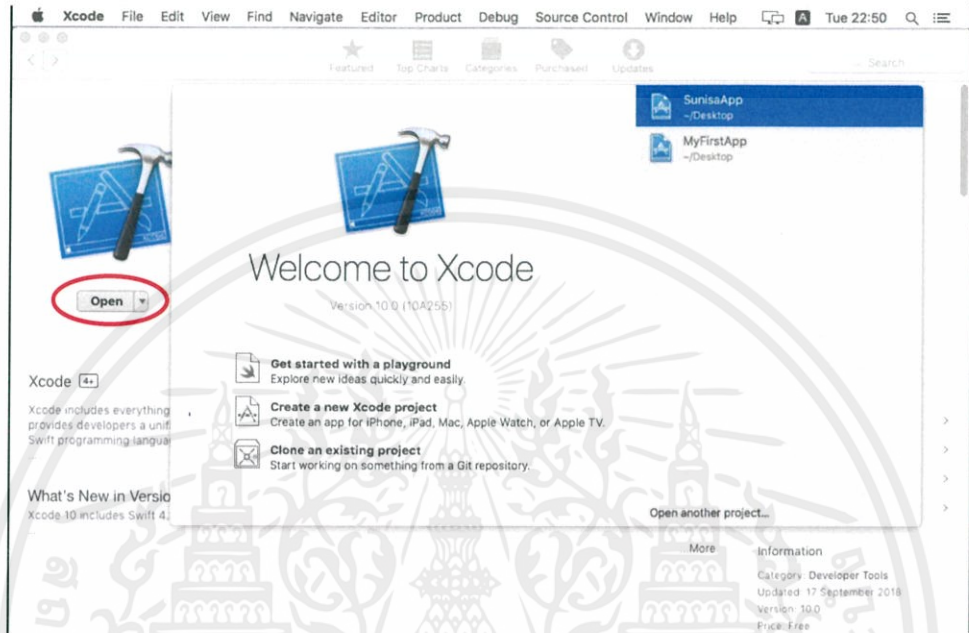


รูปที่ 2.6 กด GET และ Install

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เมื่อ Install เสร็จแล้วกด Open และจะเห็นหน้าต่างโปรแกรม Xcode ดังรูปที่

2.7



รูปที่ 2.7 หน้าต่างโปรแกรม Xcode

3) เมื่อติดตั้งโปรแกรม Xcode เสร็จแล้วให้เปิดหน้าต่างโปรแกรม Xcode ไว้ดังรูปที่ 5 และรันโปรแกรมใน Visual Studio Code จะเห็น iOS Simulator ขึ้นมาให้อัตโนมัติโดยที่ไม่ต้องสร้างขึ้นเองดังรูปที่ 2.8



รูปที่ 2.8 ตัวอย่างการจำลองโทรศัพท์มือถือ iPhone

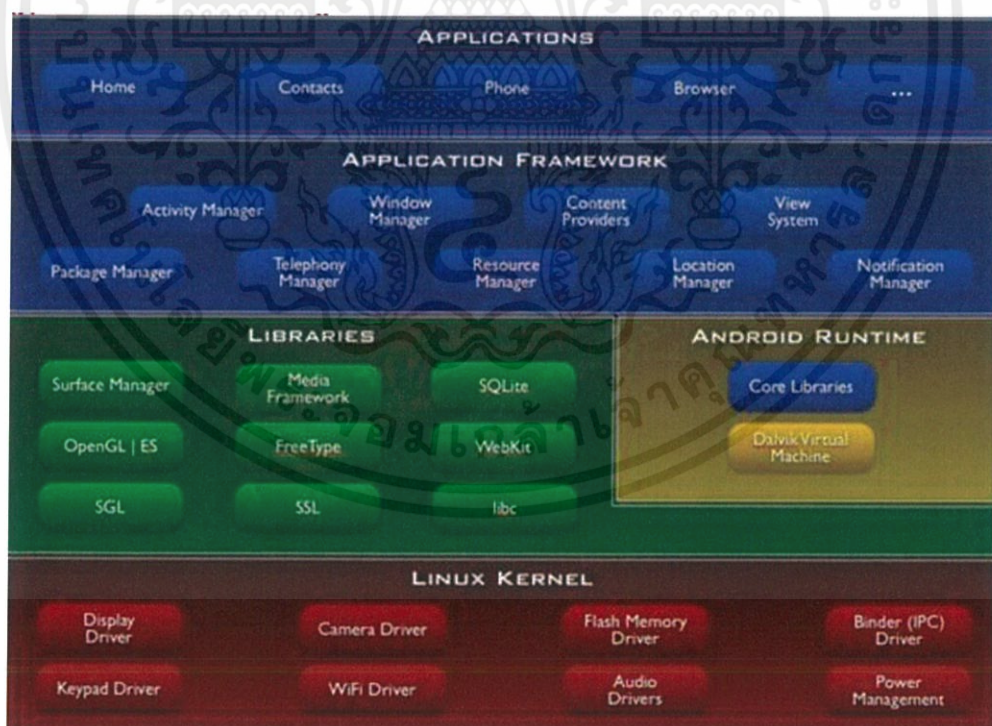
2.6 ระบบปฏิบัติการแอนดรอยด์

แอนดรอยด์ (Android) คือระบบปฏิบัติการแบบเปิดเผยซอร์สโค้ด (Open Source) โดยบริษัท กูเกิล (Google Inc.) ที่ได้รับความนิยมเป็นอย่างสูง เนื่องจากอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์มีจำนวนมาก อุปกรณ์มีหลากหลายระดับ หลายราคา สามารถทำงานบนอุปกรณ์ที่มีขนาดหน้าจอและความละเอียดแตกต่างกันได้ ทำให้ผู้บริโภคสามารถเลือกได้ตามต้องการ หากมองในทิศทางสำหรับนักพัฒนาโปรแกรม (Programmer) การพัฒนาโปรแกรมเพื่อใช้งานบนระบบปฏิบัติการแอนดรอยด์ไม่ใช่เรื่องยาก เพราะมีข้อมูลในการพัฒนารวมทั้ง Android SDK (Software Development Kit) เตรียมไว้ให้กับนักพัฒนาได้เรียนรู้ และเมื่อนักพัฒนาต้องการจะเผยแพร่หรือจำหน่ายโปรแกรมที่พัฒนาแล้วเสร็จ แอนดรอยด์ยังมีตลาดในการเผยแพร่โปรแกรมผ่าน Android Market หากกล่าวถึงโครงสร้างภาษาที่ใช้ในการพัฒนานั้น สำหรับ Android SDK จะยึดโครงสร้างของภาษาจาวา (Java language) ในการเขียนโปรแกรม เพราะโปรแกรมที่พัฒนามาได้จะต้องทำงานอยู่ภายใต้ Dalvik Virtual Machine เช่นเดียวกับโปรแกรมจาวา ที่ต้องทำงานอยู่ภายใต้ Java Virtual Machine

ระบบปฏิบัติการแอนดรอยด์ถูกพัฒนามาจากบริษัท แอนดรอยด์ (Android Inc.) เมื่อปี พ.ศ 2546 โดยมีนาย แอนดี รูบิน (Andy Rubin) ผู้ให้กำเนิดระบบปฏิบัติการนี้ และถูกบริษัท กูเกิล ซื้อกิจการเมื่อเดือนสิงหาคม ปี พ.ศ 2548 โดยบริษัท แอนดรอยด์ ได้กลายเป็นมาบริษัทลูกของบริษัท กูเกิล และยังมีนาย แอนดี รูบิน ดำเนินงานอยู่ในทีมพัฒนาระบบปฏิบัติการ

ระบบปฏิบัติการแอนดรอยด์ เป็นระบบปฏิบัติการที่พัฒนามาจากการนำเอาแกนกลางของระบบปฏิบัติการลินุกซ์ (Linux Kernel) ซึ่งเป็นระบบปฏิบัติการที่ออกแบบมาเพื่อทำงานเป็นเครื่องให้บริการ (Server) มาพัฒนาต่อ เพื่อให้กลายเป็นระบบปฏิบัติการบนอุปกรณ์พกพา (Mobile Operating System) และช่วงต่อมาได้มีการออกผลิตภัณฑ์จากบริษัทต่าง ๆ ออกมาหลากหลายรุ่น หลากหลายยี่ห้อ ตามการพัฒนาของระบบปฏิบัติการแอนดรอยด์ที่มีอยู่อย่างต่อเนื่อง ทำให้สินค้าของแอนดรอยด์มีให้เลือกอยู่อย่างมากมาย

การทำความเข้าใจโครงสร้างของระบบปฏิบัติการแอนดรอยด์ ถือว่าเป็นสิ่งสำคัญ เพราะถ้านักพัฒนาโปรแกรมสามารถมองภาพโดยรวมของระบบได้ทั้งหมดจะช่วยให้สามารถเข้าใจถึงกระบวนการทำงานได้ดียิ่งขึ้นและสามารถนำไปช่วยในการออกแบบโปรแกรมที่ต้องการพัฒนาเพื่อให้เกิดประสิทธิภาพในการทำงาน โครงสร้างของระบบปฏิบัติการแอนดรอยด์แสดงดังรูปที่ 2.9



รูปที่ 2.9 โครงสร้างของแอนดรอยด์ [6]

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์ดังรูปที่ 2.9 จะสังเกตได้ว่าการแบ่งออกเป็นส่วน ๆ ที่มีความเกี่ยวเนื่องกัน โดยส่วนบนสุดจะเป็นส่วนที่ผู้ใช้งานทำการติดต่อโดยตรง ซึ่งก็คือส่วนของแอปพลิเคชัน (Applications) จากนั้นจะลำดับลงมาเป็นองค์ประกอบอื่น ๆ ตามลำดับ และสุดท้ายจะเป็นส่วนที่ติดต่อกับอุปกรณ์โดยผ่านทาง Linux Kernel โครงสร้างของแอนดรอยด์สามารถอธิบายเป็นส่วน ๆ ได้ดังนี้

1) Applications หรือส่วนของโปรแกรมที่มีมากับระบบปฏิบัติการ หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่าง ๆ ได้โดยตรง ซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ดโปรแกรมเอาไว้

2) Application Framework เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวกและมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมาก ๆ เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งาน Application Framework ในส่วนที่ต้องการใช้งานแล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน ตัวอย่างเช่น

- Activities Manager เป็นกลุ่มของชุดคำสั่งที่จัดการเกี่ยวกับวงจรการทำงานของหน้าต่างโปรแกรม (Activity)

- Content Providers เป็นกลุ่มของชุดคำสั่ง ที่ใช้ในการเข้าถึงข้อมูลของโปรแกรมอื่นและสามารถแบ่งปันข้อมูลให้โปรแกรมอื่นเข้าถึงได้

- View System เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับการจัดการโครงสร้างของหน้าจอที่แสดงผลในส่วนที่ติดต่อกับผู้ใช้งาน (User Interface)

- Telephony Manager เป็นกลุ่มของชุดคำสั่งที่ใช้ในการเข้าถึงข้อมูลด้านโทรศัพท์ เช่น หมายเลขโทรศัพท์ เป็นต้น

- Resource Manager เป็นกลุ่มของชุดคำสั่งในการเข้าถึงข้อมูลที่เป็นข้อความ, รูปภาพ

- Location Manager เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับตำแหน่งทางภูมิศาสตร์ ที่ระบบปฏิบัติการได้รับค่าจากอุปกรณ์

- Notification Manager เป็นกลุ่มของชุดคำสั่งที่จะถูกเรียกใช้เมื่อโปรแกรมต้องการแสดงผลให้กับผู้ใช้งานผ่านทางแถบสถานะ (Status Bar) ของหน้าจอ

3) Libraries เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย C/C++ โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตามวัตถุประสงค์ของการใช้งาน เช่น Surface Manage จัดการเกี่ยวกับการแสดงผล, Media

Framework จัดการเกี่ยวกับการการการแสดงภาพและเสียง, Open GL | ES และ SGL จัดการเกี่ยวกับภาพ 3 มิติ และ 2 มิติ, SQLite จัดการเกี่ยวกับระบบฐานข้อมูล เป็นต้น

4) Android Runtime จะมี Dalvik Virtual Machine ที่ถูกออกแบบมา เพื่อให้ทำงานบนอุปกรณ์ที่มีหน่วยความจำ(Memory), หน่วยประมวลผลกลาง(CPU) และพลังงาน (Battery)ที่จำกัด ซึ่งการทำงานของ Dalvik Virtual Machine จะทำการแปลงไฟล์ที่ต้องการทำงานไปเป็นไฟล์ .DEX ก่อนการทำงาน เหตุผลก็เพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับหน่วยประมวลผลกลางที่มีความเร็วไม่มาก ส่วนต่อมาคือ Core Libraries ที่เป็นส่วนรวบรวมคำสั่งและชุดคำสั่งสำคัญ โดยถูกเขียนด้วยภาษาจาวา (Java Language)

5) Linux Kernel เป็นส่วนที่ทำหน้าที่หัวใจสำคัญในการจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่าง ๆ ความปลอดภัย เครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจากระบบปฏิบัติการลินุกซ์รุ่น 2.6 (Linux 2.6. Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี [6]

2.7 ระบบปฏิบัติการไอโอเอส

ระบบปฏิบัติการไอโอเอส (iOS) มีชื่อเดิมว่า iPhone OS เริ่มต้นด้วยการเปิดตัวของ iPhone เมื่อวันที่ 29 มิถุนายน 2550 ระบบปฏิบัติการไอโอเอส (iOS) เป็นระบบปฏิบัติการสำหรับสมาร์ทโฟน (Smartphone) ของ Apple โดยเริ่มต้นพัฒนาสำหรับใช้ในโทรศัพท์ iPhone และได้พัฒนาต่อใช้สำหรับ iPod Touch และ iPad โดยระบบปฏิบัติการนี้สามารถเชื่อมต่อไปยัง App Store สำหรับการเข้าถึงถึงแอปพลิเคชัน (Application) Apple ได้มีการพัฒนาปรับปรุงสำหรับ iPhone, iPad และ iPod Touch ผ่านทางระบบ iTunes คือโปรแกรมฟรี สำหรับ Mac และ PC ใช้ดูหนังฟังเพลงบนคอมพิวเตอร์ รวมทั้งจัดระเบียบและ Sync ทุก ๆ อย่าง เป็นร้านขายความบันเทิงบนคอมพิวเตอร์, บน iPod touch, iPhone และ iPad ที่มีทุก ๆ อย่าง ในทุกที่และทุกเวลา พัฒนาระบบรักษาความปลอดภัยให้มีความเป็นเลิศ ซึ่งนี่คือข้อได้เปรียบเมื่อเทียบกับคู่แข่ง

ข้อดีของระบบปฏิบัติการไอโอเอส คือ มี Application หลากหลาย มีบริการ App Store และโปรแกรม iTunes สนับสนุนการจัดการอุปกรณ์มีเมนูการใช้งาน รวดเร็วและเข้าใจง่าย โปรแกรม Web Browser (Safari) ตอบสนองได้รวดเร็ว

ข้อเสียของระบบปฏิบัติการไอโอเอส คือ ผู้ใช้งานไม่สามารถออกแบบปรับเปลี่ยนหน้าจอได้ตามความต้องการ ไม่สามารถทำงานได้พร้อม ๆ กันหลายอย่าง เช่น ไม่สามารถฟังเพลงพร้อมเปิด Web Browser เพื่อใช้งานอินเทอร์เน็ตได้ [7]

2.8 ระบบฐานข้อมูล

ฐานข้อมูล หรือ Database คือกลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ในที่เดียวกันอย่างเป็นระบบเพื่อให้สามารถค้นหา เพิ่มเติม ลบ และแก้ไขข้อมูลได้อย่างสะดวกและมีประสิทธิภาพ ซึ่งข้อดีของการใช้งานฐานข้อมูลเมื่อเทียบกับไฟล์ได้แก่

- 1) การเข้าถึงข้อมูลในฐานข้อมูลมีความรวดเร็วกว่าการเข้าถึงข้อมูลในไฟล์
- 2) การค้นหาข้อมูลที่มีเงื่อนไขตามที่กำหนดฐานข้อมูลสามารถทำได้ง่ายกว่าไฟล์
- 3) มีคุณสมบัติรองรับการเข้าถึงข้อมูลพร้อมกัน (concurrent access)
- 4) มีการรักษาความถูกต้องและความน่าเชื่อถือของข้อมูล
- 5) การจัดเก็บข้อมูลอยู่ที่เดียวกัน ทำให้ใช้งานข้อมูลร่วมกันได้

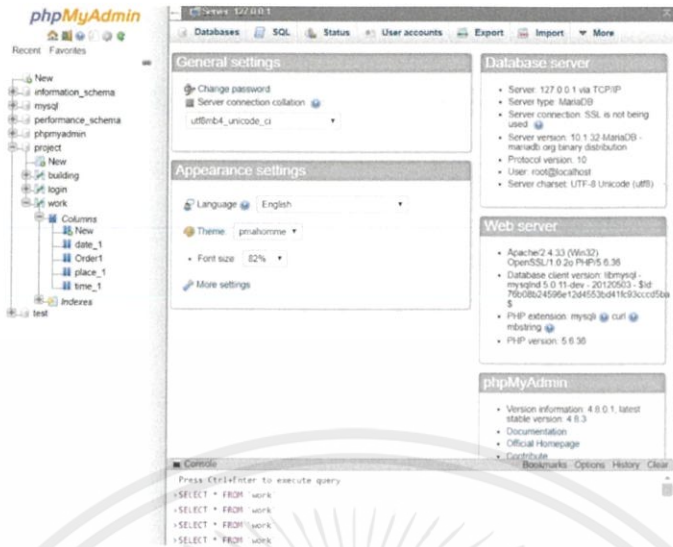
ในการสร้างเว็บเพจที่เรียบง่ายและไม่มีการเปลี่ยนแปลงบ่อย ๆ การใช้ภาษา HTML เพียงอย่างเดียวอาจเพียงพอต่อการใช้งาน แต่ในปัจจุบันความต้องการรับรู้ข่าวสารต่าง ๆ ให้รวดเร็ว และทันต่อเหตุการณ์ จึงจำเป็นต้องนำเทคโนโลยีอื่น ๆ เข้ามาช่วยให้สามารถปรับปรุงข้อมูลได้ทันที่ และลดความผิดพลาดที่อาจเกิดขึ้นจากการที่ต้องปฏิบัติงานกับข้อมูลจำนวนมากและสิ่งหนึ่งที่ถูกนำมาใช้เพื่อแก้ปัญหาเหล่านี้คือ ระบบฐานข้อมูลบนเว็บ (Web Database)

เมื่อมีความจำเป็นต้องนำระบบจัดการฐานข้อมูลมาใช้ในเว็บไซด์ที่เขียนขึ้นด้วยภาษา PHP ระบบจัดการฐานข้อมูลที่มีมักจะถูกเลือกใช้เป็นลำดับต้น ๆ คือ MySQL ด้วยคุณสมบัติเด่นหลายประการ ทำให้ผู้ใช้ PHP ส่วนใหญ่เลือกใช้ MySQL เป็นระบบจัดการฐานข้อมูลสำหรับเว็บไซด์และ Web Application

ในโครงการนี้ได้เลือกใช้ MySQL ร่วมกับ phpMyAdmin ในการจัดการฐานข้อมูล ซึ่งมีหน้าตาของโปรแกรมดังรูปที่ 2.10

2.8.1 MySQL

MySQL คือ โปรแกรมระบบจัดการฐานข้อมูลที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูลที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่น เพื่อให้ได้ระบบงานที่รองรับความต้องการของผู้ใช้ เช่น ทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) เช่น ภาษา PHP ภาษา asp.net ภาษา JSP เป็นต้น หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น ภาษา Visual Basic.net ภาษา Java หรือภาษา C# เป็นต้น โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลโอเพนซอร์ซ (Open Source) ที่ถูกนำไปใช้งานมากที่สุด [8]

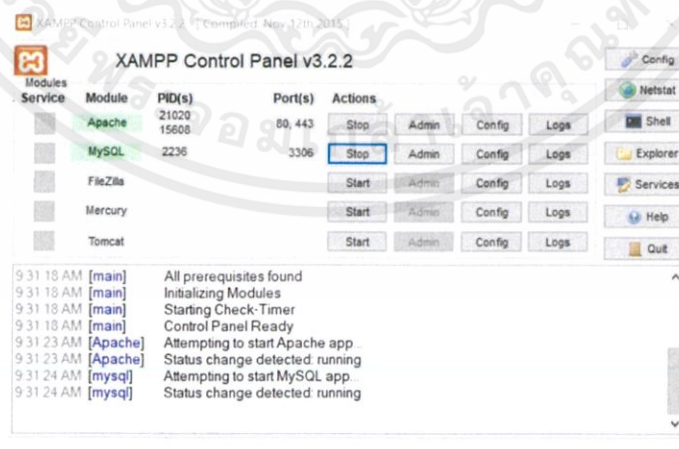


รูปที่ 2.10 โปรแกรมจัดการฐานข้อมูล MySQL

2.8.2 โปรแกรม XAMPP

XAMPP เป็นโปรแกรมไว้จำลอง Web server เพื่อไว้ทดสอบสคริปหรือเว็บไซต์ในเครื่องของเรา โดยที่ไม่ต้องเชื่อมต่ออินเทอร์เน็ตและไม่ต้องมีค่าใช้จ่ายใด ๆ ง่ายต่อการติดตั้งและใช้งาน โปรแกรม XAMPP จะมาพร้อมกับ PHP ภาษาสำหรับพัฒนา Web Application ที่เป็นที่ยอมรับ, โปรแกรมฐานข้อมูล (MySQL), โปรแกรม Apache และภาษา Perl อีกทั้งยังมาพร้อมกับ OpenSSL และ phpMyAdmin

โปรแกรม XAMPP มีหน้าต่างของโปรแกรมดังรูปที่ 2.11 สามารถใช้งานได้ 4 OS ได้แก่ วินโดวส์ (Windows), ลินุกซ์ (Linux), แมคโอเอส (Mac OS), โซลาริส (Solaris) [9]



รูปที่ 2.11 โปรแกรม XAMPP

2.8.3 ภาษา PHP

PHP คือภาษาคอมพิวเตอร์จำพวก Scripting language คำสั่งต่าง ๆ จะเก็บอยู่ในไฟล์ที่เรียกว่า Script และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่ง ซึ่ง PHP ได้รับการพัฒนาและออกแบบมาเพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า Server-side หรือ HTML-embedded scripting language และ PHP เป็นภาษาที่สามารถนำมาทำเว็บเพจและสามารถนำข้อมูลจาก Database Server มาแสดงในเว็บเพจได้ [10]

2.9 Marvel App

ในการออกแบบหน้าตาของแอปพลิเคชันในโครงการนี้ได้เลือกใช้ Marvel App ซึ่งเป็นเครื่องมือที่ช่วยออกแบบ UX/UI สำหรับเว็บไซต์หรือ Mobile app และ Marvel App สามารถเก็บข้อมูลที่ออกแบบหรือ Prototype ที่ทำการออกแบบนั้นไว้บน Cloud ที่รองรับทั้ง Google drive หรือ Dropbox [11] มีหน้าตาของแอปพลิเคชันดังรูปที่ 2.12



รูปที่ 2.12 Marvel App ออกแบบหน้าตาสำหรับแอปพลิเคชัน

2.10 บาร์โค้ด (Barcode)

บาร์โค้ด (Barcode) หรือในภาษาไทยเรียกว่า “รหัสแท่ง” ประกอบด้วยเส้นมืด (มักจะเป็นสีดำ) และเส้นสว่าง (มักจะเป็นสีขาว) วางเรียงกันเป็นรหัสแทนตัวเลขและตัวอักษร ใช้เพื่ออำนวยความสะดวกให้เครื่องคอมพิวเตอร์สามารถอ่านรหัสข้อมูลได้ง่ายขึ้น โดยใช้เครื่องอ่านบาร์โค้ด (Barcode Scanner) ซึ่งจะทำงานได้รวดเร็วและช่วยลดความผิดพลาดในการคีย์ข้อมูลได้

โดยหลักการบาร์โค้ดจะถูกอ่านด้วยเครื่องสแกนเนอร์บันทึกข้อมูลเข้าไปเก็บในคอมพิวเตอร์โดยตรงไม่ต้องกดปุ่มที่แทนพิมพ์ ทำให้มีความสะดวก รวดเร็วในการทำงานรวมถึงอ่านข้อมูลได้อย่างถูกต้องแม่นยำ เชื่อถือได้และจะเห็นได้ชัดเจนว่าปัจจุบันระบบบาร์โค้ดเข้าไปมีบทบาทในทุกส่วนของอุตสาหกรรมการค้าขายและการบริการ ที่ต้องใช้ในการบริหารจัดการข้อมูลจากฐานข้อมูลในคอมพิวเตอร์ ปัจจุบันมีการประยุกต์การใช้งานบาร์โค้ดเข้ากับการใช้งานของโทรศัพท์มือถือสามารถพกพาได้สะดวก เพื่อทำการจัดเก็บแสดงผล ตรวจสอบและประมวลในด้านอื่น ๆ ได้ด้วย

2.10.1 ข้อดีของบาร์โค้ด

- 1) ช่วยเพิ่มประสิทธิภาพในการทำงานระบบบาร์โค้ดจะช่วยให้การทำงานเป็นไปอย่างรวดเร็ว แม่นยำและมีประสิทธิภาพในการเก็บข้อมูล
- 2) ลดขั้นตอนในการทำงาน ประหยัดเวลา โดยเฉพาะการรับชำระเงิน การออกใบเสร็จ ระบบสินค้าคงคลัง ซึ่งระบบบาร์โค้ดจะตัดยอดสินค้าโดยอัตโนมัติ สามารถให้ข้อมูลเกี่ยวกับการหมุนเวียนสินค้าได้ดี
- 3) ลดข้อผิดพลาดในการทำธุรกิจ การใช้บาร์โค้ดในการทำธุรกิจการค้าเป็นการป้องกันการเกิดความผิดพลาดต่าง ๆ ในระบบธุรกิจมากขึ้น
- 4) ลดค่าใช้จ่าย ลดค่าใช้จ่ายในส่วนต่าง ๆ ในการจ้างพนักงานเช็คสต็อกสินค้าหรือทำหน้าที่อื่น ๆ ที่บาร์โค้ดสามารถทำแทนได้
- 5) ระดับมาตรฐานสินค้า การระบุบาร์โค้ดจะเป็นการแสดงข้อมูลสินค้าของผู้ผลิต และทำให้ผู้ผลิตคำนึงถึงการปรับปรุงคุณภาพสินค้าอย่างต่อเนื่องเพื่อให้สอดคล้องกับมาตรฐานการผลิตภัณฑ์อุตสาหกรรมเรื่องการแสดงข้อมูลสินค้าและเพิ่มความน่าเชื่อถือให้กับลูกค้าอีกด้วย

2.10.2 ข้อเสียของบาร์โค้ด

- 1) เสียหายง่าย แถบบาร์โค้ดเสียหายได้ง่าย เพียงแค่มีรอยเปื้อนสกปรก แถบสี หรือสีจางไปเมื่อถูกแสงแดด หรือความชื้น
- 2) เครื่องอ่านบาร์โค้ดมีข้อจำกัดในการทำงาน เมื่อนำเครื่องอ่านบาร์โค้ดไปใช้งานในสภาพแวดล้อมที่เปียกชื้น คลื่นแสงที่ใช้ในการอ่านจะถูกหักเหง่าย เมื่อแถบบาร์โค้ดมีการเปียกชื้นด้วยสาเหตุดังกล่าวทำให้การอ่านข้อมูลในแถบบาร์โค้ดผิดพลาดได้
- 3) ขณะการอ่านแถบบาร์โค้ด จำเป็นอย่างยิ่งที่จะต้องเห็นแถบบาร์โค้ด หากแถบบาร์โค้ดถูกปิดบัง ทำให้ไม่สามารถที่จะอ่านข้อมูลได้
- 4) ความเร็ว เครื่องอ่านบาร์โค้ดไม่สามารถที่จะอ่านแถบบาร์โค้ดที่เคลื่อนที่ด้วยความรวดเร็ว ดังนั้นหากแถบบาร์โค้ดติดอยู่บนวัตถุที่เคลื่อนที่ด้วยความรวดเร็วจะมีผลทำให้ความแม่นยำในการอ่านต่ำลง [12]

2.10.3 ประเภทของบาร์โค้ด

2.9.3.1 มาตรฐานบาร์โค้ด 1 มิติ มีหลายแบบ ซึ่งมีดังต่อไปนี้

- 1) 2 of 5 (non-interleaved) ปัจจุบันพบน้อย แต่ยังมีการใช้อยู่บ้าง เช่น ตัวสายการบิน ห้องแล็บภาพ ซึ่งเป็นโค้ดที่ใช้ในการเข้ารหัสข้อมูลที่เก่าแก่ที่สุด 2 of 5 (non-interleaved) และ 2 of 5 (interleaved) หลายคนมักสับสนเพราะ 2 of 5 (interleaved) ยังมีความนิยมแพร่หลายอยู่ซึ่งถือว่าเป็นคนละโค้ดกันและเครื่องมือหรือโปรแกรมอาจจะไม่สนับสนุน ดังนั้นต้องดูให้แน่ใจว่าเป็น non-interleaved หรือ interleaved
- 2) Bookland เป็นบาร์โค้ดรุ่นเก่าที่ใช้กับพวกหนังสือ ปัจจุบันถูกแทนที่ด้วย ISBN-13 เป็นโค้ดค้าปลีกพวกหนังสือ นิตยสารหรือผลิตภัณฑ์อื่นที่เกี่ยวข้องกับหนังสือ โดยมีส่วนประกอบจาก EAN13 และ Supplemental code 5 ตัว ซึ่งจะแสดง ID ของหนังสือและราคาปก แสดงดังรูปที่ 2.13

ISBN 978-3-16-148410-0



รูปที่ 2.13 ISBN-13 [13]

3) Codabar เป็นบาร์โค้ด ที่ใช้ใน FED-EX ห้างสมุด ธนาคาร
เลือด แสดงดังรูปที่ 2.14



รูปที่ 2.14 Codabar [13]

4) Code128 จัดเป็นโค้ดที่มีความนิยมมาก มันมีความหนาแน่น
กะทัดรัด การป้อนข้อมูลที่ดีกว่า Code39 และ Code128 สามารถสนันสนุนได้ 128 ASCII ตัว
(ทั้งตัวเลขและอักษร) แสดงดังรูปที่ 2.15



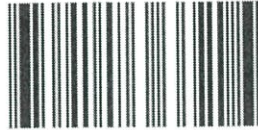
รูปที่ 2.15 Code128 [13]

5) Code39 เป็นโค้ดรุ่นเก่า แต่ยังมีคามนิยมในการใช้งานอยู่
ใช้กับงาน Inventory และตรวจติดตาม สามารถบรรจุได้ทั้งตัวเลขและตัวอักษร สามารถพิมพ์ได้
หลายขนาด โดยแบบพื้นฐานจะรองรับอักษร A-Z, 0-9 แสดงดังรูปที่ 2.16



รูปที่ 2.16 Code39 [13]

6) Code93 เป็นโค้ดที่ไม่ค่อยมีการใช้งาน ขนาดกะทัดรัด ใช้ใน
งานชิ้นส่วนอิเล็กทรอนิกส์และเนื่องจากชื่อที่คล้ายคลึงระหว่าง Code93 และ Code39 ซึ่งทำให้
เข้าใจผิดได้ ดังนั้นหากมีการใช้งานในโค้ดดังกล่าว จะตัดชื่ออุปกรณ์หรือซอฟต์แวร์ต้องแน่ใจว่าตรง
กับที่จะใช้งาน แสดงดังรูปที่ 2.17



CODE93

รูปที่ 2.17 Code93 [13]

7) EAN ย่อมาจาก European Article Number นิยมใช้กับสินค้าปลีกในสหรัฐ ญี่ปุ่น และแพร่หลายในอุตสาหกรรมค้าปลีกทั่วโลก แสดงดังรูปที่ 2.18



รูปที่ 2.18 EAN Code [13]

8) GS1-128 (EAN-128) เป็นโค้ดพิเศษของ Code128 ที่มีการถอดรหัสกับระบบ GS1 แสดงดังรูปที่ 2.19



รูปที่ 2.19 GS1-128 (EAN-128) [13]

9) Interleaved 2 of 5 (ITF) เป็นโค้ดที่เป็นตัวเลขเท่านั้น มีขนาดเล็กเพราะมีการเข้ารหัสทั้งในแถบและช่องว่างใช้ในกล่องกระดาษลูกฟูกในอุตสาหกรรมขนส่งสินค้า แสดงดังรูปที่ 2.20



รูปที่ 2.20 Interleaved 2 of 5 [13]

10) ISSN ใช้ในอุตสาหกรรมงานพิมพ์ วารสารอิเล็กทรอนิกส์

แสดงดังรูปที่ 2.21



รูปที่ 2.21 ISSN [13]

11) MSI Plessey ตัวเลขสัญลักษณ์ ที่ใช้ในห้องสมุด แสดงดังรูป

ที่ 2.22



รูปที่ 2.22 MSI Plessey [13]

12) POSTNET บาร์โค้ดที่ใช้ในการเข้ารหัสไปรษณีย์ของ
ไปรษณีย์สหรัฐ ซึ่งมีแตกต่างจากบาร์โค้ดอื่น ๆ มีความพิเศษที่แตกต่างคือ จะมีความสูงของแท่งไม่
เท่ากันและระยะห่างระหว่างแท่งเท่ากัน ซึ่งบรรจุข้อมูลของรหัสไปรษณีย์และพื้นที่จัดส่ง แสดงดัง
รูปที่ 2.23



รูปที่ 2.23 POSTNET Barcode [13]

13) UPC ย่อมาจาก Universal Product Code ใช้มากกับสินค้า
ปลีกในสหรัฐ แคนาดา อังกฤษ ออสเตรเลีย และนิวซีแลนด์ แสดงดังรูปที่ 2.24



รูปที่ 2.24 UPC Barcode [13]

14) Supplemental barcode เป็นโค้ดที่เพิ่มเติมทางด้านขวา พบใน UPC, EAN หรือ ISBN-13 จะเป็นตัวอักษรที่เพิ่มมา 2-5 ตัว [13]

2.9.3.1 มาตรฐานบาร์โค้ด 2 มิติ มีหลายแบบ ซึ่งมีดังต่อไปนี้

1) QR Code QR (Quick Response) ประดิษฐ์คิดค้นโดย บริษัท เด็นโซเวฟ จัดเป็นบาร์โค้ดชนิดที่เครื่องอ่าน 2D ทั่วไปทุกยี่ห้อสามารถอ่านได้ และแอปพลิเคชันบนโทรศัพท์มือถือหลายตัวสามารถอ่านได้ ซึ่งเคยพบสิ่งพิมพ์ลงในสื่อต่าง ๆ เมื่อสแกนมาก็เป็นที่อยู่เว็บ มีความหนาแน่นสูง ทำให้บรรจุอักษรได้มาบนพื้นที่เล็ก ๆ ขนาดจะเป็นสี่เหลี่ยมจัตุรัส ทำให้พื้นที่น้อยและไม่ต้องมีรูปแบบในการบรรจุข้อมูล สามารถใส่ได้ตามต้องการ แสดงดังรูปที่ 2.25



รูปที่ 2.25 QR Code [13]

2) Maxi Code Barcode ประกอบด้วย จุดเหมือนรังผึ้งรอบ ๆ กระจาย วงกลมตรงกลาง พบในการใช้ส่งพัสดุของสหรัฐ ต้องใส่ข้อมูลลงตามรูปแบบที่กำหนด แสดงดังรูปที่ 2.26



รูปที่ 2.26 Maxi Code [13]

3) PDF417 คิดค้นโดย Symbol Technologies บรรจุได้ถึง ประมาณ 1100 ตัว มีลักษณะเป็นสี่เหลี่ยมผืนผ้า มีแถบเริ่มต้นและแถบปิดท้ายทำให้กินพื้นที่มากขึ้นกว่าโค้ดอื่น ๆ ไม่ต้องมีรูปแบบในการบรรจุข้อมูล สามารถใส่ได้ตามต้องการ แสดงดังรูปที่ 2.27



รูปที่ 2.27 PDF417 [13]

4) Data Matrix หากดูผิวเผินจะคล้ายกับ QR Code แต่ไม่เหมือนกัน การอ่านและถอดรหัสจะทำในเส้นทแยงมุม (QR Code จะอ่านตามแนวตรง) พบในผลิตภัณฑ์จากทางสหรัฐและยุโรป จุดอ่อนคือ ต้องใส่ข้อมูลลงตามรูปแบบที่กำหนด [13] แสดงดังรูปที่ 2.28



รูปที่ 2.28 Data Matrix [13]

2.11 Google Maps API

Google Maps คือ บริการแผนที่ของ Google ซึ่งให้บริการ Services ที่เกี่ยวข้องกับแผนที่ทั้งหมด โดยในปัจจุบันแผนที่ของ Google นั้นมีอยู่หลากหลายประเภท เช่น ที่ให้บริการแผนที่บนเว็บไซต์หรือแอปพลิเคชันบนโทรศัพท์มือถือและทางช่องทางอื่น ๆ โดย Services เหล่านี้สามารถเรียกใช้งานได้ฟรี ในกรณีที่ผ่านแอปพลิเคชันทั่ว ๆ ไป แต่ถ้าในกรณีที่มีการเรียกใช้งานในเว็บไซต์หรือแอปพลิเคชันที่พัฒนาขึ้น Google Maps จะมี API ให้ใช้งานได้ แต่ Services ต่าง ๆ ของ Google นั้นมีข้อจำกัดในการใช้งาน แต่หากต้องการใช้ในปริมาณสูงขึ้นไปจะต้องซื้อ Package ที่ทาง Google Maps มีมาให้ ซึ่งโดยปกติจะมีการจำกัดจำนวนที่ Request เข้ามาเรียกใช้งาน [14]

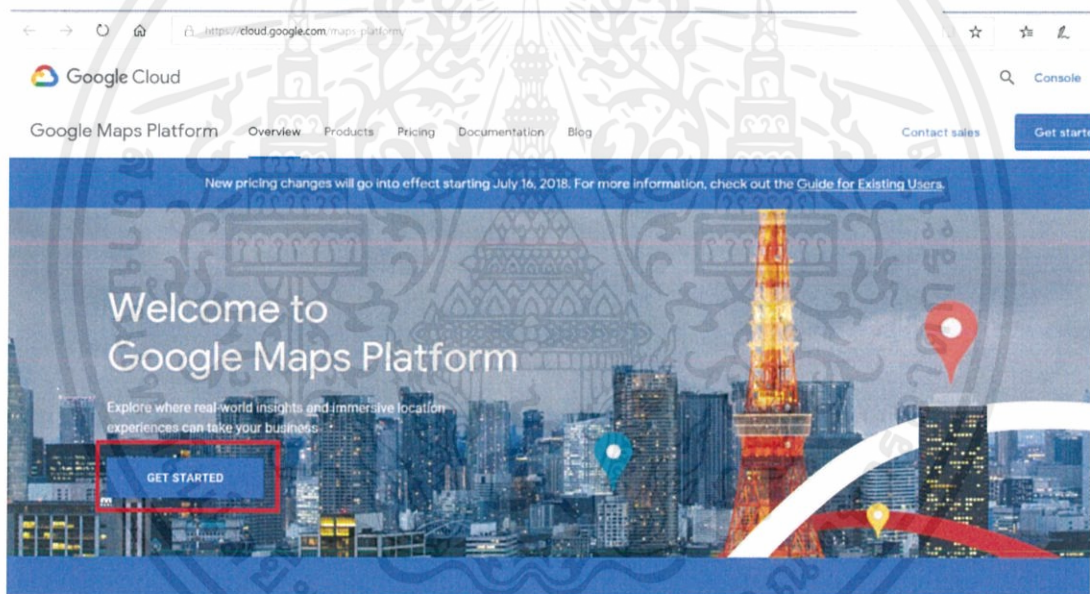
Google Maps API เป็นชุด API ของ Google สำหรับพัฒนา Web Application และ Mobile Application ไว้สำหรับเรียกใช้แผนที่และชุด Services ต่าง ๆ ของ Google เพื่อพัฒนา Application ได้เหมือนกับ Google โดยแผนที่ยังมี features ต่าง ๆ ให้เรียกใช้งาน เช่น ดังต่อไปนี้

- 1) การปรับแต่งแผนที่ (Styled Maps)
- 2) ชุดควบคุมแผนที่ (Maps Control)
- 3) ชุดเครื่องมือวาดภาพบนแผนที่ (Drawing)

- 4) การนำทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง (Directions Service)
- 5) การคำนวณความสูงของจุดพิกัด (Elevation Service)
- 6) การแปลงที่อยู่เป็นพิกัด Latitude และ Longitude (Geo Coding Service)
- 7) การดึงข้อมูล POI (Point of Interest) คือ ข้อมูลสถานที่ต่าง ๆ ที่ Google รวบรวมไว้ให้ เช่น โรงแรม ห้างสรรพสินค้า โรงเรียน สถานที่ราชการต่าง ๆ และอื่นๆ อีกมากมาย (Places API) มาใช้งานในแอปพลิเคชัน [15]

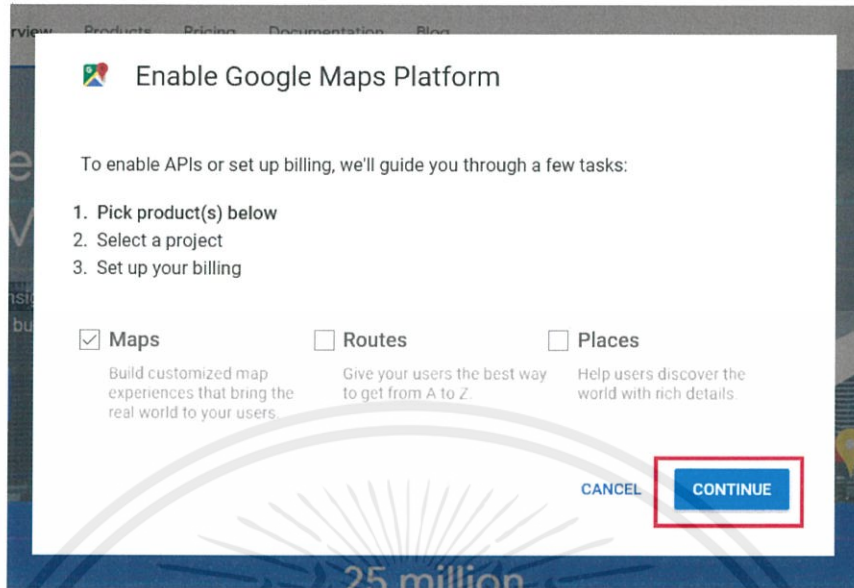
วิธีการขอ Google Maps API Key สำหรับใช้งาน Google Maps มีดังต่อไปนี้

- 1) เข้าไปที่ <https://cloud.google.com/maps-platform> [16]
- 2) คลิกที่ GET STARTED ดังรูปที่ 2.29



รูปที่ 2.29 กด GET STARTED

- 3) เลือกการใช้งาน Maps และกด CONTINUE ดังรูปที่ 2.30

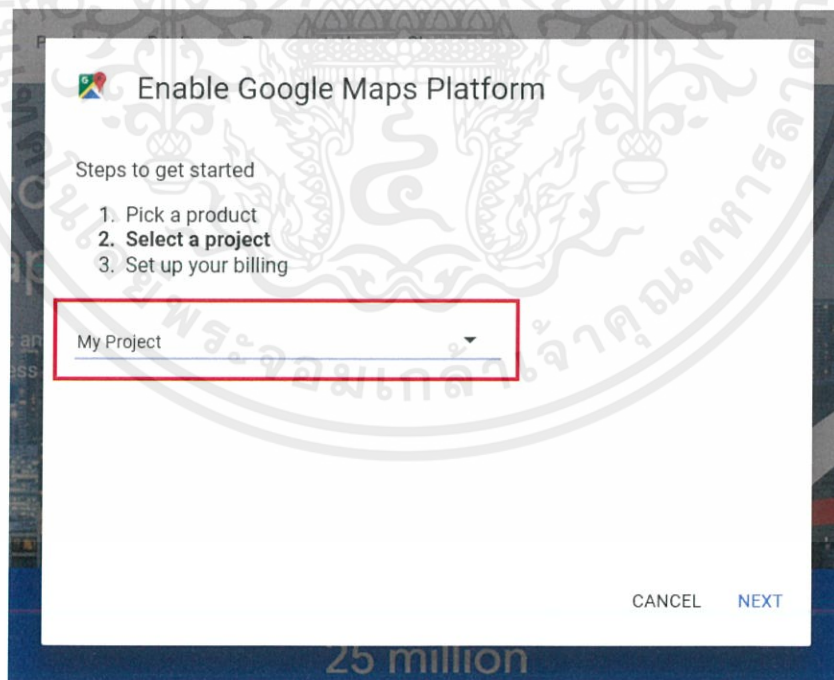


รูปที่ 2.30 เลือก Maps

4) หากยังไม่ได้ Login ให้เข้าใช้งานด้วยบัญชีของ Gmail ก่อน

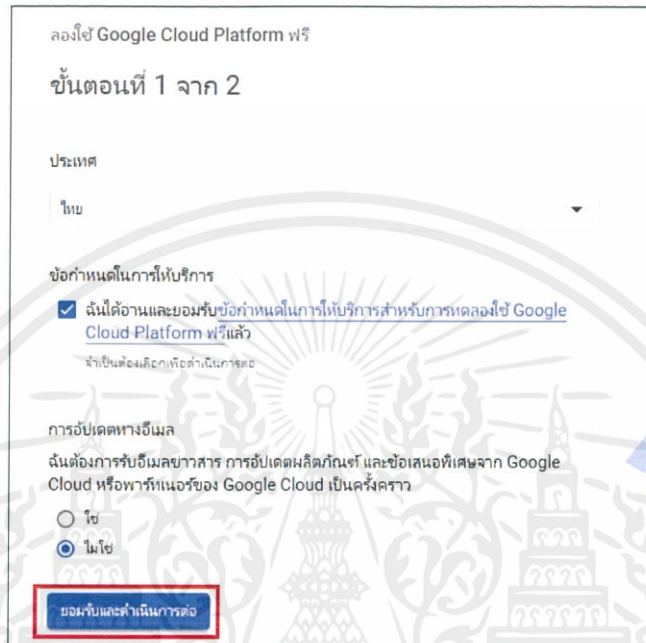
5) ตั้งชื่อ Project ตามที่ต้องการ ชื่อที่ตั้งต้องมีตัวอักษรมากกว่า 4 ตัวขึ้นไป และกด

NEXT ดังรูปที่ 2.31



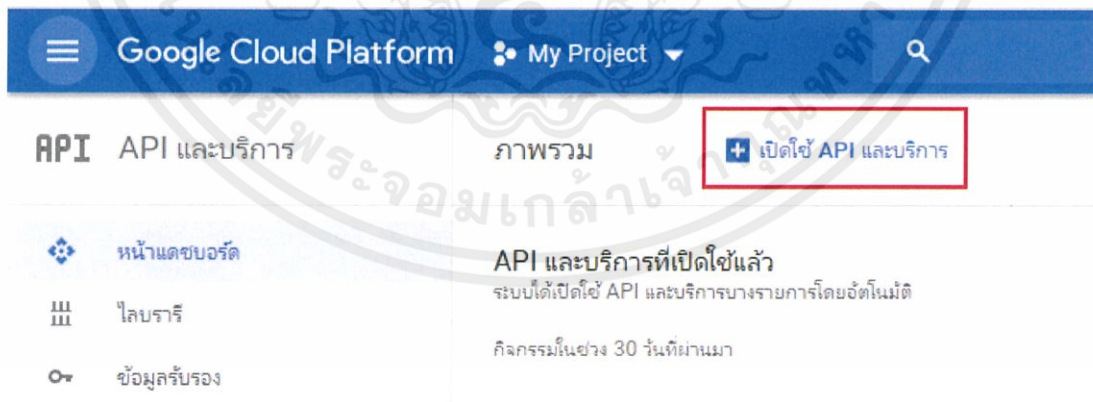
รูปที่ 2.31 ตั้งชื่อ Project

6) ขึ้นหน้าต่างดังรูปที่ 2.32 ให้เลือกประเทศและยอมรับเพื่อทดลองใช้ Google Cloud Platform ฟรี จากนั้นกด “ยอมรับและดำเนินการต่อ” และขึ้นมาให้กรอกข้อมูลต่อ ให้กรอกข้อมูลให้ครบ



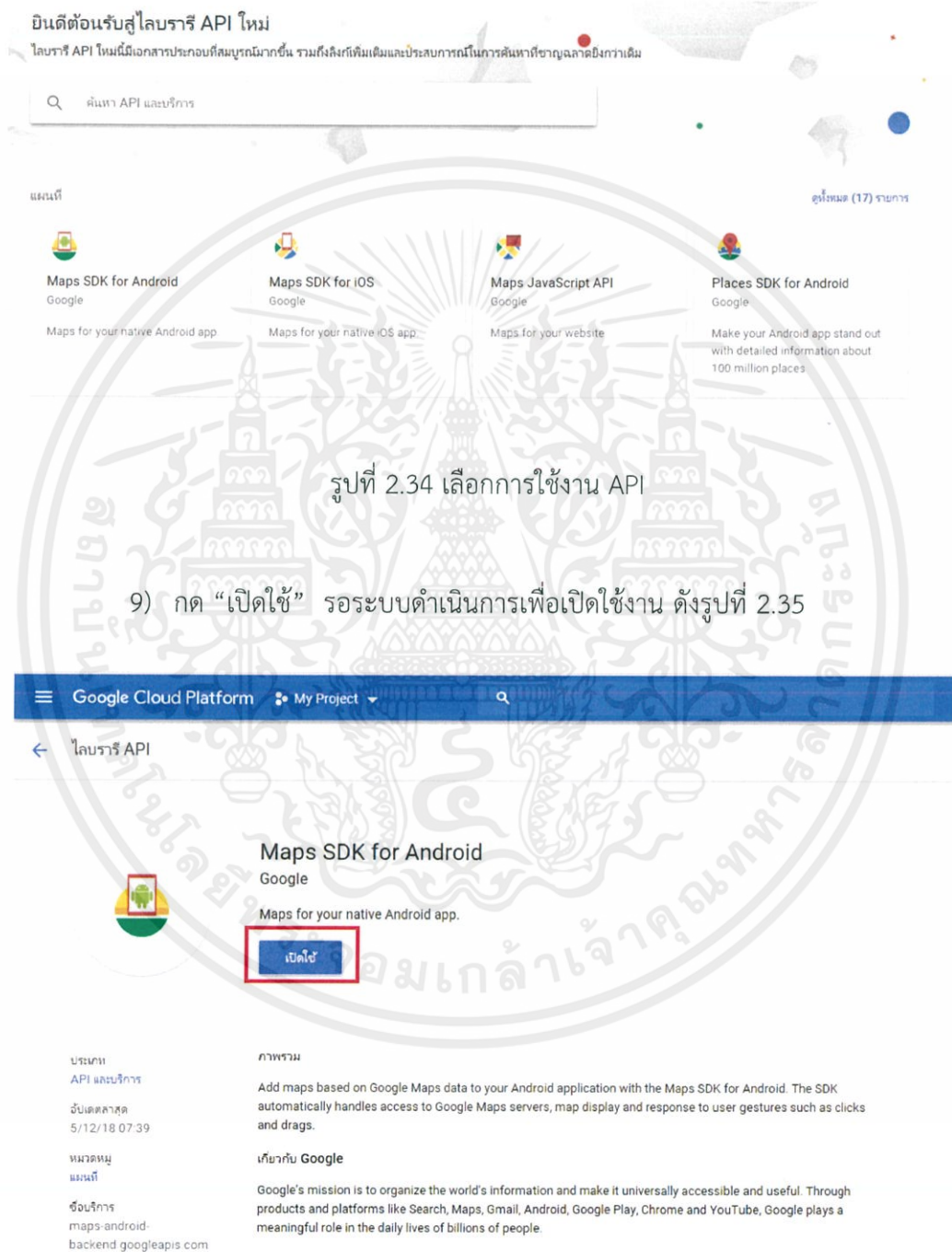
รูปที่ 2.32 กรอกข้อมูล

7) ระบบจะพามาที่หน้าต่างดังรูปที่ 2.33 กด “เปิดใช้ API และบริการ”



รูปที่ 2.33 เปิดใช้ API และบริการ

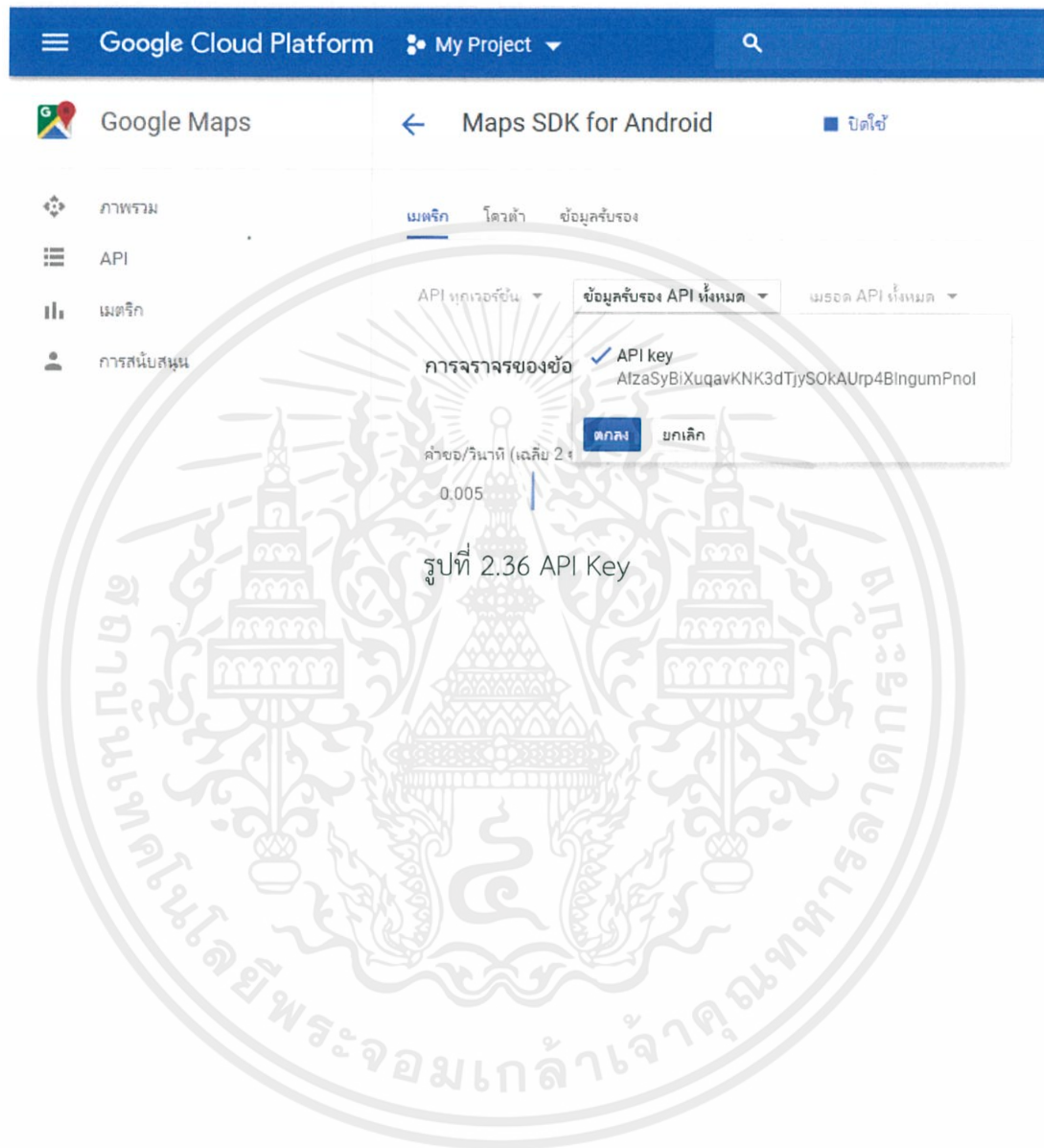
8) เลือกเพื่อเปิดใช้งาน API ดังรูปที่ 2.34 ยกตัวอย่างเลือก “Maps SDK for Android”



รูปที่ 2.35 เปิดการใช้งาน API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10) จากนั้นกด API และเลือกการใช้งานที่เปิด API ระบบแสดงหน้าจอตั้งรูปที่ 2.36 และสามารถนำ API key ดังกล่าวไปใช้งานต่อไป



บทที่ 3

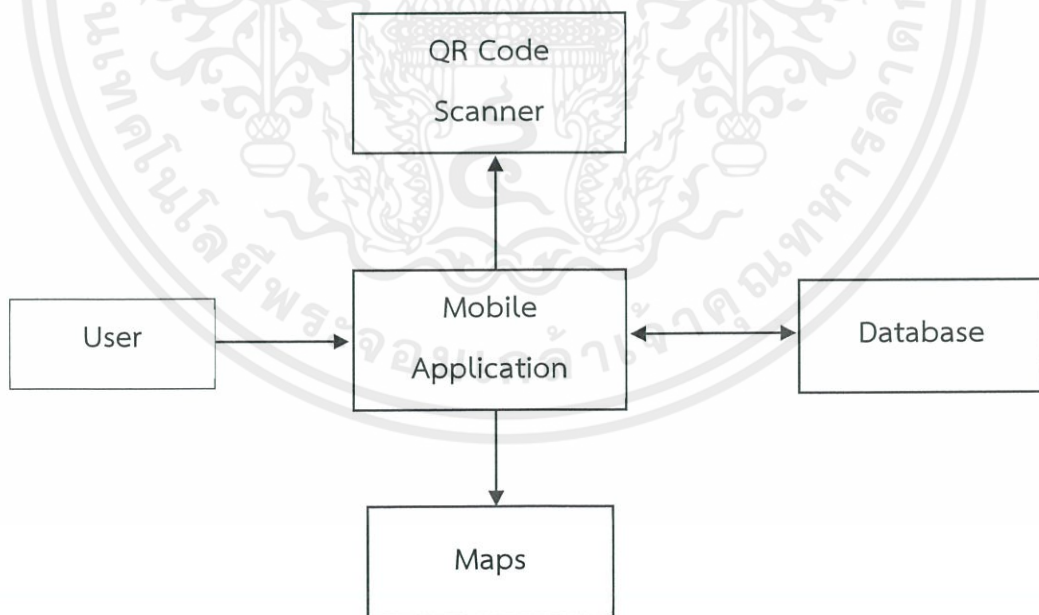
วิธีดำเนินการโครงการ

การออกแบบและพัฒนาระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบินของบริษัท สามารถคอมเทค จำกัด ในสนามบินสุวรรณภูมิ มีวัตถุประสงค์เพื่ออำนวยความสะดวกให้กับพนักงานในบริษัท โดยการจัดทำโครงการและออกแบบการทำงานของระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบินมีดังต่อไปนี้

3.1 การออกแบบ

3.1.1 การออกแบบการทำงานของระบบโดยรวม

โครงการนี้ต้องการพัฒนาระบบการทำงานในบริษัท สามารถคอมเทค จำกัด ซึ่งในโครงการนี้ได้จัดทำแอปพลิเคชันบนโทรศัพท์มือถือ มีการใช้งาน 3 ระบบคือ (1) ระบบการทำงานการบำรุงรักษาเชิงป้องกัน (Preventive Maintenance) (2) ระบบการทำงานการบำรุงรักษาเชิงแก้ไข (Corrective Maintenance) (3) ระบบคลังพัสดุ (Store) โดยบล็อกไดอะแกรมของระบบแสดงดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบินของบริษัท สามารถคอมเทค จำกัด ในสนามบินสุวรรณภูมิ

เมื่อผู้ใช้งาน (User) ได้เข้ามาในระบบ “บริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบิน” สามารถเลือกการใช้งานตามความต้องการของผู้ใช้ ระบบสามารถให้ผู้ใช้งาน (User) ได้ปฏิบัติงานสะดวกยิ่งขึ้นโดยสามารถออกแบบฟอร์มต่าง ๆ ผ่านบนโทรศัพท์มือถือและสามารถนำข้อมูลที่กรอกไว้ดังกล่าวออกมาใช้งานต่อได้ ได้แก่ แบบฟอร์มการบำรุงรักษาเครือข่าย แบบฟอร์มการนำเข้าและนำออกของคลังพัสดุ แบบฟอร์มการแจ้งเปลี่ยนอุปกรณ์ แบบฟอร์มกรอกปัญหาที่พบของอุปกรณ์เครือข่าย โดยการกรอกแบบฟอร์มสามารถใช้ฟังก์ชันการอ่านคิวอาร์โค้ดผ่านโทรศัพท์มือถือได้ การกรอกข้อมูลดังกล่าวจะถูกส่งไปที่ฐานข้อมูล และระบบได้ดึงข้อมูลจากฐานข้อมูลมาแสดง ได้แก่ แสดงข้อมูลรายละเอียดของอุปกรณ์เครือข่ายตามสถานที่ต่าง ๆ ข้อมูลในคลังพัสดุ ข้อมูลปัญหาที่เคยขึ้นกับอุปกรณ์เครือข่าย ข้อมูลรายละเอียดการนำเข้าและนำออกของคลังพัสดุ และระบบสามารถนำทางไปสถานที่ต่าง ๆ ที่มีอุปกรณ์เครือข่าย โดยใช้ Google Map ในการนำทาง

3.1.2 การออกแบบและสร้างตารางฐานข้อมูล

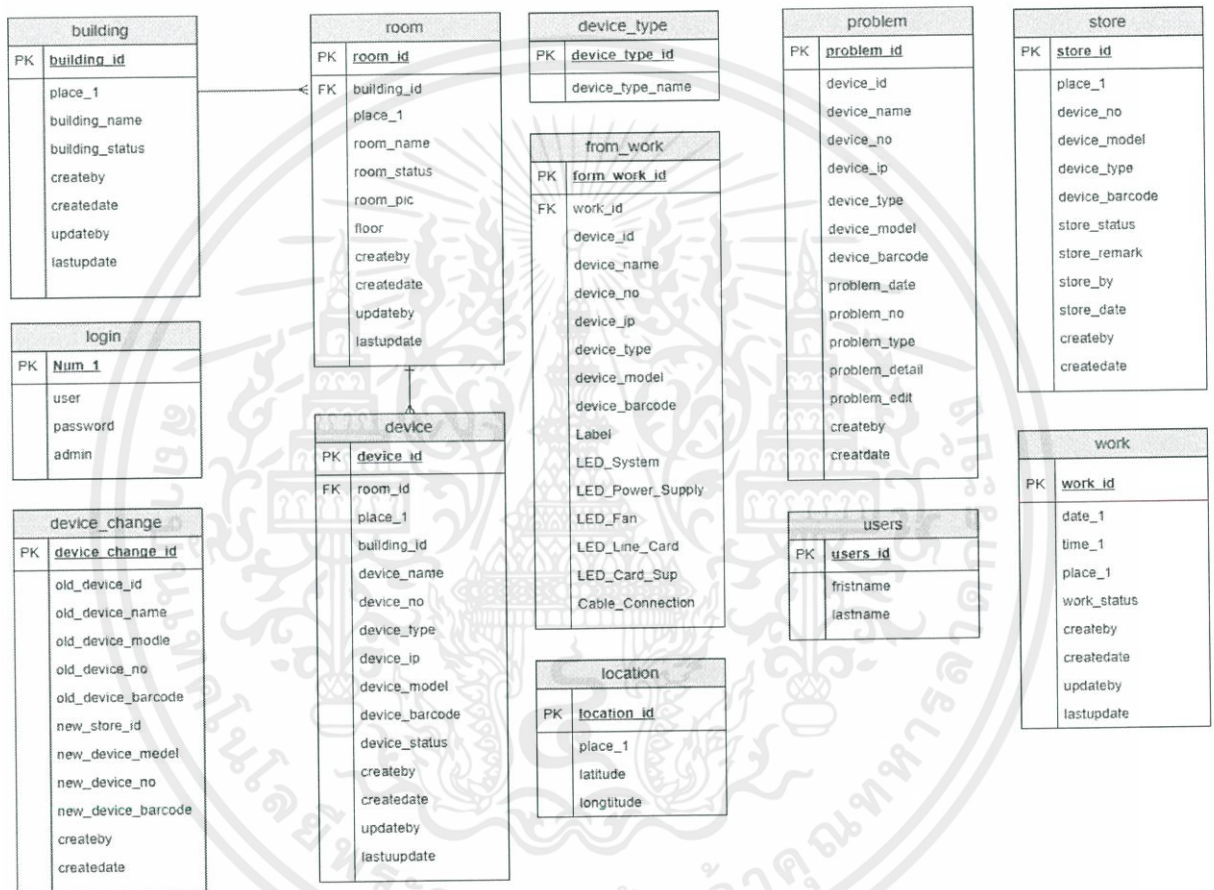
ในการสร้างฐานข้อมูล ผู้จัดทำได้ออกแบบฐานข้อมูลสำหรับจัดเก็บข้อมูลระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบินของบริษัท สามารถคอมเทค จำกัด ในสนามบินสุวรรณภูมิแสดงในรูปที่ 3.2 โดยทำการออกแบบตารางทั้งหมด 12 ตารางได้แก่

- 1) ตาราง “building” ใช้เก็บข้อมูลสถานที่ ชื่ออาคารที่ตั้งอยู่ในสถานที่นั้น
- 2) ตาราง “device” ใช้เก็บข้อมูลชื่อของอุปกรณ์, ชนิด, เลขไอพีแอดเดรส, โมเดล, หมายเลขเครื่อง และเลขครุภัณฑ์ของอุปกรณ์ที่ถูกติดตั้งในแต่ละสถานที่
- 3) ตาราง “device_change” ใช้สำหรับเก็บข้อมูลของอุปกรณ์ที่ถูกเปลี่ยนจากอุปกรณ์เดิมเป็นอุปกรณ์ใหม่
- 4) ตาราง “device_type” ใช้สำหรับการเก็บข้อมูลชนิดของอุปกรณ์ต่าง ๆ
- 5) ตาราง “form_work” ใช้สำหรับเก็บค่าสถานะของอุปกรณ์ในการทำแบบฟอร์มการบำรุงรักษาเครือข่าย
- 6) ตาราง “location” ใช้สำหรับเก็บค่าพิกัดละติจูดและค่าลองจิจูดของสถานที่ และใช้ในการปัดมุดใน map
- 7) ตาราง “login” ใช้สำหรับเก็บค่า username และ password สำหรับการเข้า login ในระบบ
- 8) ตาราง “problem” ใช้สำหรับเก็บข้อมูลปัญหาที่เกิดขึ้นกับอุปกรณ์ต่างๆในสถานที่ต่างๆ วันเดือนปีที่เกิดปัญหา และวิธีแก้ไขปัญหา
- 9) ตาราง “room” ใช้สำหรับเก็บข้อมูล ชื่อห้อง และชื่อรูปภาพของหน้าห้อง

10) ตาราง “store” ใช้สำหรับข้อมูล ใช้สำหรับเก็บข้อมูลของอุปกรณ์ที่ถูกนำเข้า – นำออก ในคลังพัสดุ

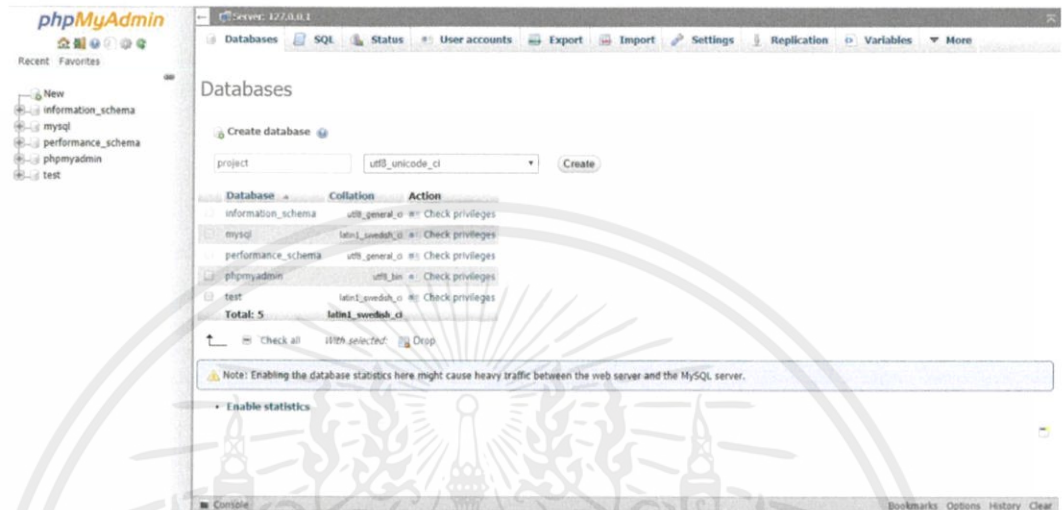
11) ตาราง “user” ใช้สำหรับเก็บชื่อ – นามสกุล ไว้สำหรับเลือกในการนำอุปกรณ์เข้า – ออกภายในคลังพัสดุ

12) ตาราง “work” ใช้สำหรับเก็บข้อมูลตารางงาน PM โดยประกอบไปด้วย สถานที่ วันเดือนปี และเวลา ในการเข้าปฏิบัติงาน



รูปที่ 3.2 แผนภาพการออกแบบฐานข้อมูล

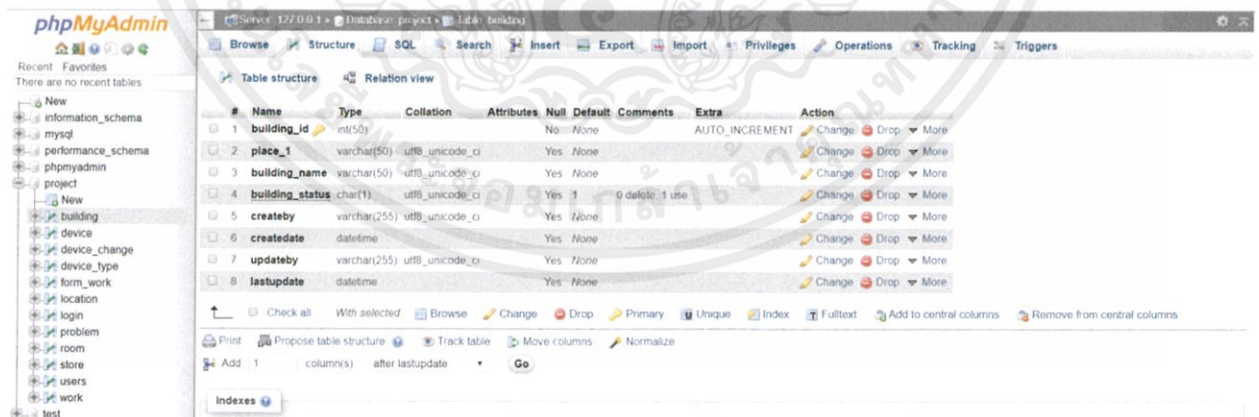
เมื่อทำการออกแบบฐานข้อมูลระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบินเรียบร้อยแล้ว ผู้จัดทำได้ใช้โปรแกรม phpMyAdmin ในการสร้างฐานข้อมูล โดยตั้งชื่อฐานข้อมูลว่า project และกำหนด collection เป็น utf8_general_ci เพื่อให้สามารถรองรับข้อมูลที่เป็นภาษาไทยได้ ดังแสดงในรูปที่ 3.3



รูปที่ 3.3 การสร้างฐานข้อมูล

จากนั้นสร้างตารางในฐานข้อมูลทั้งหมด 12 ตารางดังต่อไปนี้

1) ตาราง “building” ประกอบไปด้วยข้อมูลของอาคารมีทั้งหมด 8 คอลัมน์ ได้แก่ building_id, place_1, building_name, building_status, createby, createdate, updateby และ lastupdate ดังแสดงในรูปที่ 3.4



รูปที่ 3.4 ตาราง “building”

2) ตาราง “device” ประกอบไปด้วยข้อมูลของอุปกรณ์มีทั้งหมด 15 คอลัมน์ ได้แก่ device_id, place_1, building_id, room_id, device_name, device_no, device_type, device_ip, device_model, device_barcode, device_status, createby, createdate, updateby และ lastupdate ดังแสดงในรูปที่ 3.5

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	device_id	int(50)			No	None		AUTO_INCREMENT	Change Drop More
2	place_1	varchar(50)	utf8_unicode_ci		Yes	None			Change Drop More
3	building_id	int(11)			Yes	None			Change Drop More
4	room_id	int(50)			Yes	None			Change Drop More
5	device_name	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
6	device_no	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
7	device_type	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
8	device_ip	varchar(50)	utf8_unicode_ci		Yes	None			Change Drop More
9	device_model	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
10	device_barcode	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
11	device_status	char(1)	utf8_unicode_ci		Yes	1	0 delete 1 use		Change Drop More
12	createby	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
13	createdate	datetime			Yes	None			Change Drop More
14	updateby	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
15	lastupdate	datetime			Yes	None			Change Drop More

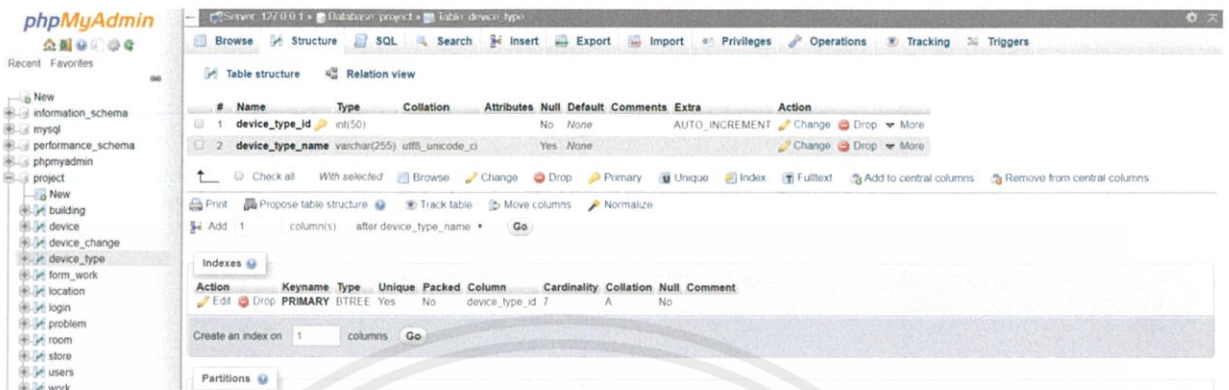
รูปที่ 3.5 ตาราง “device”

3) ตาราง “device_change” ประกอบไปด้วยข้อมูลที่แสดงการเปลี่ยนอุปกรณ์เดิมเป็นอุปกรณ์ใหม่มีทั้งหมด 12 คอลัมน์ ได้แก่ device_change_id, old_device_id, createdate, old_device_name, old_device_no, old_device_barcode, new_store_id, createby, new_device_model และ new_device_barcode ดังแสดงในรูปที่ 3.6

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	device_change_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	old_device_id	int(255)			Yes	None			Change Drop More
3	old_device_name	varchar(255)	utf8_general_c		Yes	None			Change Drop More
4	old_device_model	varchar(255)	utf8_general_c		Yes	None			Change Drop More
5	old_device_no	varchar(255)	utf8_general_c		Yes	None			Change Drop More
6	old_device_barcode	varchar(255)	utf8_general_c		Yes	None			Change Drop More
7	new_store_id	int(11)			Yes	None			Change Drop More
8	new_device_model	varchar(255)	utf8_general_c		Yes	None			Change Drop More
9	new_device_no	varchar(255)	utf8_general_c		Yes	None			Change Drop More
10	new_device_barcode	varchar(255)	utf8_general_c		Yes	None			Change Drop More
11	createby	varchar(255)	utf8_general_c		Yes	None			Change Drop More
12	createdate	datetime			Yes	None			Change Drop More

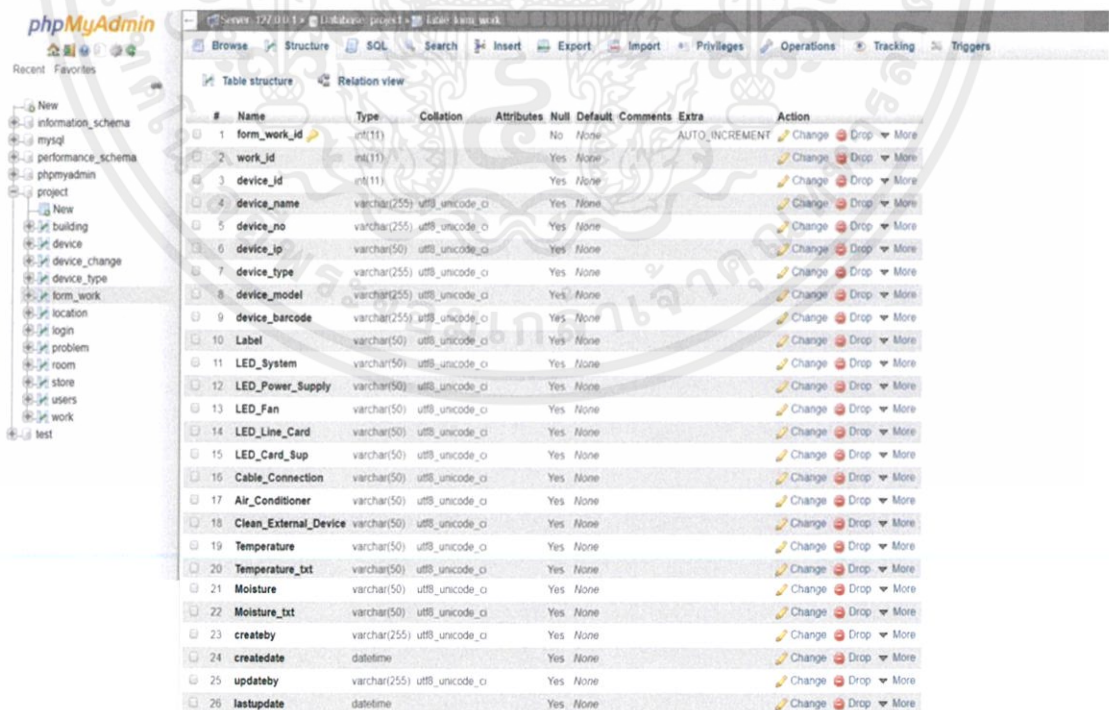
รูปที่ 3.6 ตาราง “device_change”

4) ตาราง “device_type” ประกอบไปด้วยข้อมูลประเภทอุปกรณ์มีทั้งหมด 2 คอลัมน์ ได้แก่ device_type_id และ device_type_name ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 ตาราง “device_type”

5) ตาราง “form_work” ประกอบไปด้วยข้อมูลค่าสถานะของอุปกรณ์ที่ได้ทำการตรวจสอบมีทั้งหมด 26 คอลัมน์ ได้แก่ form_work_id, work_id, device_name, device_no, device_ip, device_type, device_model, device_barcode, Label, LED_System, LED_Power_Supply, LED_Fan, LED_Line_Card, LED_Card_Sup, Cable_Connection, Air_Conditioner, Clean_External_Device, Temperature, Teperature_txt, Moisture, Moisture_txt, createby, createdate, updateby และ lastupdate ดังแสดงในรูปที่ 3.8



รูปที่ 3.8 ตาราง “form_work”

6) ตาราง “location” ประกอบไปด้วยข้อมูลสถานที่ที่มีทั้งหมด 4 คอลัมน์ ได้แก่ locotion_id, place_1, latitude และ longtitude ดังแสดงในรูปที่ 3.9

The screenshot shows the phpMyAdmin interface for a database named 'project'. The 'Table structure' tab is selected for the 'location' table. The table has four columns: 'location_id' (int(11), PRIMARY, AUTO_INCREMENT), 'place_1' (varchar(50)), 'latitude' (varchar(255)), and 'longtitude' (varchar(255)). All columns are of type 'utf8_unicode_ci' and are not nullable. The primary key is 'location_id'.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	location_id	int(11)	utf8_unicode_ci		No	None		AUTO_INCREMENT	Change Drop More
2	place_1	varchar(50)	utf8_unicode_ci		No	None			Change Drop More
3	latitude	varchar(255)	utf8_unicode_ci		No	None			Change Drop More
4	longtitude	varchar(255)	utf8_unicode_ci		No	None			Change Drop More

รูปที่ 3.9 ตาราง “location”

7) ตาราง “login” ประกอบไปข้อมูลของผู้เข้าใช้งานมีทั้งหมด 4 คอลัมน์ ได้แก่ user, password, Num_1 และ admin ดังแสดงในรูปที่ 3.10

The screenshot shows the phpMyAdmin interface for a database named 'project'. The 'Table structure' tab is selected for the 'login' table. The table has four columns: 'user' (varchar(40)), 'password' (varchar(40)), 'Num_1' (int(40)), and 'admin' (char(1)). 'Num_1' is the primary key and has the AUTO_INCREMENT attribute. All columns are of type 'utf8_unicode_ci' and are not nullable.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user	varchar(40)	utf8_unicode_ci		No	None			Change Drop More
2	password	varchar(40)	utf8_unicode_ci		No	None			Change Drop More
3	Num_1	int(40)	utf8_unicode_ci		No	None		AUTO_INCREMENT	Change Drop More
4	admin	char(1)	utf8_unicode_ci		Yes	None			Change Drop More

รูปที่ 3.10 ตาราง “login”

8) ตาราง “problem” ประกอบไปด้วยข้อมูลปัญหาที่เกิดขึ้นกับอุปกรณ์มี 15 คอลัมน์ ได้แก่ problem_id, device_id, device_name, device_no, device_ip, device_type, device_model, device_barcode, problem_date, problem_no, problem_type, problem_detail, problem_edit, createby และ createdate ดังแสดงในรูปที่ 3.11

The screenshot shows the phpMyAdmin interface with the 'Table structure' view selected for the 'problem' table. The table has 15 columns with the following details:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	problem_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	device_id	int(11)			Yes	None			Change Drop More
3	device_name	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
4	device_no	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
5	device_ip	varchar(50)	utf8_unicode_ci		Yes	None			Change Drop More
6	device_type	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
7	device_model	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
8	device_barcode	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
9	problem_date	date			Yes	None			Change Drop More
10	problem_no	varchar(100)	utf8_unicode_ci		Yes	None			Change Drop More
11	problem_type	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
12	problem_detail	text	utf8_unicode_ci		Yes	None			Change Drop More
13	problem_edit	text	utf8_unicode_ci		Yes	None			Change Drop More
14	createby	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
15	createdate	datetime			Yes	None			Change Drop More

รูปที่ 3.11 ตาราง “problem”

9) ตาราง “room” ประกอบไปด้วยข้อมูลของห้องมีทั้งหมด 11 คอลัมน์ ได้แก่ room_id, place_1, building_id, room_name, room_status, room_pic, floor, createby, createdate, updateby และ lastupdate ดังแสดงในรูปที่ 3.12

The screenshot shows the phpMyAdmin interface with the 'Table structure' view selected for the 'room' table. The table has 11 columns with the following details:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	room_id	int(50)			No	None		AUTO_INCREMENT	Change Drop More
2	place_1	varchar(50)	utf8_unicode_ci		Yes	None			Change Drop More
3	building_id	int(11)			Yes	None			Change Drop More
4	room_name	varchar(50)	utf8_unicode_ci		Yes	None			Change Drop More
5	room_status	char(1)	utf8_unicode_ci		Yes	1	0 delete 1 use		Change Drop More
6	room_pic	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
7	floor	varchar(50)	utf8_unicode_ci		Yes	None			Change Drop More
8	createby	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
9	createdate	datetime			Yes	None			Change Drop More
10	updateby	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
11	lastupdate	datetime			Yes	None			Change Drop More

รูปที่ 3.12 ตาราง “room”

10) ตาราง “store” ประกอบไปด้วยข้อมูลของคลังพัสดุมีทั้งหมด 12 คอลัมน์ ได้แก่ store_id, place_1, device_no, device_model, device_type, device_barcode, store_status, store_ramark, store_by, store_date, createby และ creatdate ดังแสดงในรูปที่ 3.13

Table structure for 'store' table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	store_id	int(50)			No	None		AUTO_INCREMENT	Change Drop More
2	place_1	varchar(50)	utf8_unicode_ci		Yes	None			Change Drop More
3	device_no	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
4	device_model	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
5	device_type	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
6	device_barcode	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
7	store_status	char(1)	utf8_unicode_ci		Yes	1	0 output 1 input 2 in-out		Change Drop More
8	store_remark	text	utf8_unicode_ci		Yes	None			Change Drop More
9	store_by	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
10	store_date	date			Yes	None			Change Drop More
11	createby	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
12	createdate	datetime			Yes	None			Change Drop More

รูปที่ 3.13 ตาราง “store”

11) ตาราง “user” ประกอบไปด้วยข้อมูลของพนักงานมีทั้งหมด 3 คอลัมน์ ได้แก่ users_id, firstname และ lastname ดังแสดงในรูปที่ 3.14

Table structure for 'user' table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	users_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	firstname	varchar(100)	utf8_general_ci		Yes	None			Change Drop More
3	lastname	varchar(100)	utf8_general_ci		Yes	None			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	users_id	2	A	No	

รูปที่ 3.14 ตาราง “user”

12) ตาราง “work” ประกอบไปด้วยข้อมูลของตารางงานมีทั้งหมด 9 คอลัมน์ ได้แก่ work_id, date_1, time_1, place_1, work_status, createby, updateby และ lastupdate ดังแสดงในรูปที่ 3.15

Table structure for 'work' table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	work_id	int(50)			No	None		AUTO_INCREMENT	Change Drop More
2	date_1	date			No	None			Change Drop More
3	time_1	time			No	None			Change Drop More
4	place_1	varchar(50)	utf8_unicode_ci		No	None			Change Drop More
5	work_status	char(1)	utf8_unicode_ci		Yes	1	0 delete 1 waiting 2 close work		Change Drop More
6	createby	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
7	createdate	datetime			Yes	None			Change Drop More
8	updateby	varchar(255)	utf8_unicode_ci		Yes	None			Change Drop More
9	lastupdate	datetime			Yes	None			Change Drop More

รูปที่ 3.15 ตาราง “work”

3.1.3 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface) ของแอปพลิเคชัน

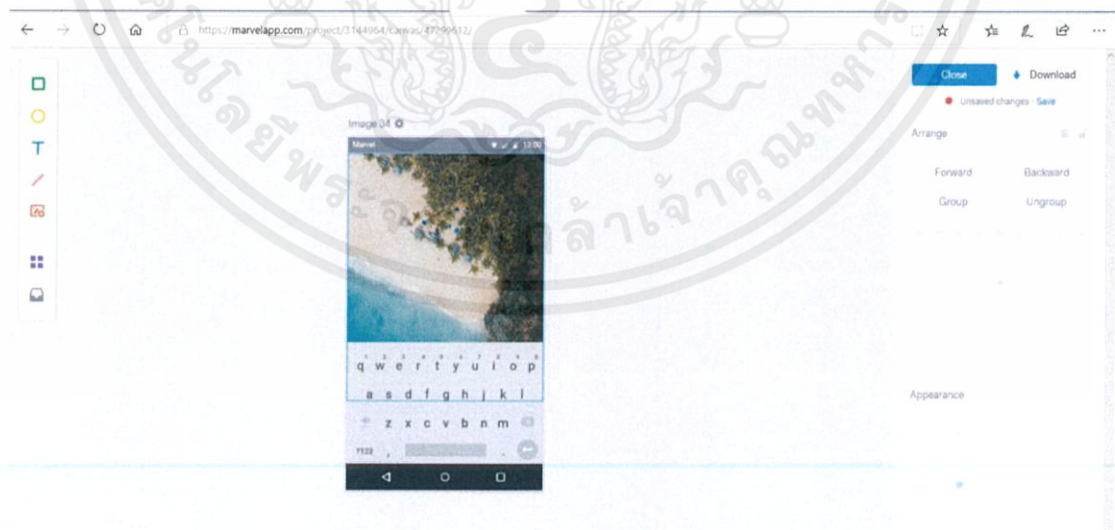
ในการออกแบบส่วนต่อประสานกับผู้ใช้ของแอปพลิเคชันในการทำงานของภาษาคอมพิวเตอร์ชนิดต่าง ๆ ที่ถูกนำมาใช้ในการออกแบบ ประกอบด้วย 2 ส่วน

1) ส่วนเบื้องหน้า (Front-End) หรือส่วนติดต่อกับผู้ใช้งาน โดยจะแสดงผลหน้าแอปพลิเคชันบนหน้าจอโทรศัพท์ ซึ่งผู้จัดทำได้ใช้ React Native โดยใช้ภาษา JavaScript ในการออกแบบและสร้างในส่วนนี้

2) ส่วนเบื้องหลัง (Back-End) หรือส่วนประมวลผลข้อมูล โดยการที่จะเข้าใช้งานในส่วนนี้ จะต้องมีการติดตั้งการใช้งานในส่วนของฐานข้อมูลเซิร์ฟเวอร์ด้วย Mysql และเซิร์ฟเวอร์ด้วย Xampp ซึ่งในส่วนนี้จะประกอบไปด้วยภาษาคอมพิวเตอร์ที่ใช้ในการออกแบบ ได้แก่ PHP และ SQL

จากนั้นทำการใช้รูปแบบการแลกเปลี่ยนข้อมูล (API : Applications Program Interface) โดยใช้ในรูปแบบ json (JavaScript Object Notation) ในการแลกเปลี่ยนข้อมูลกันระหว่างส่วนเบื้องหน้าและส่วนเบื้องหลัง

ในการออกแบบหน้าตาของแอปพลิเคชันในโครงการนี้ได้เลือกใช้ Marvel App ซึ่งเป็นเครื่องมือที่ช่วยออกแบบ UX/UI สำหรับเว็บไซต์หรือ Mobile app และ Marvel App สามารถเก็บข้อมูลที่ออกแบบหรือ Prototype ที่ทำการออกแบบนั้นไว้บน Cloud ที่รองรับทั้ง Google drive หรือ Dropbox มีหน้าตาของแอปพลิเคชันดังรูปที่ 3.16



รูปที่ 3.16 Marvel App ออกแบบหน้าตาสำหรับแอปพลิเคชัน

3.1.3.1 ออกแบบหน้าตาต่างแอปพลิเคชันสำหรับ Login

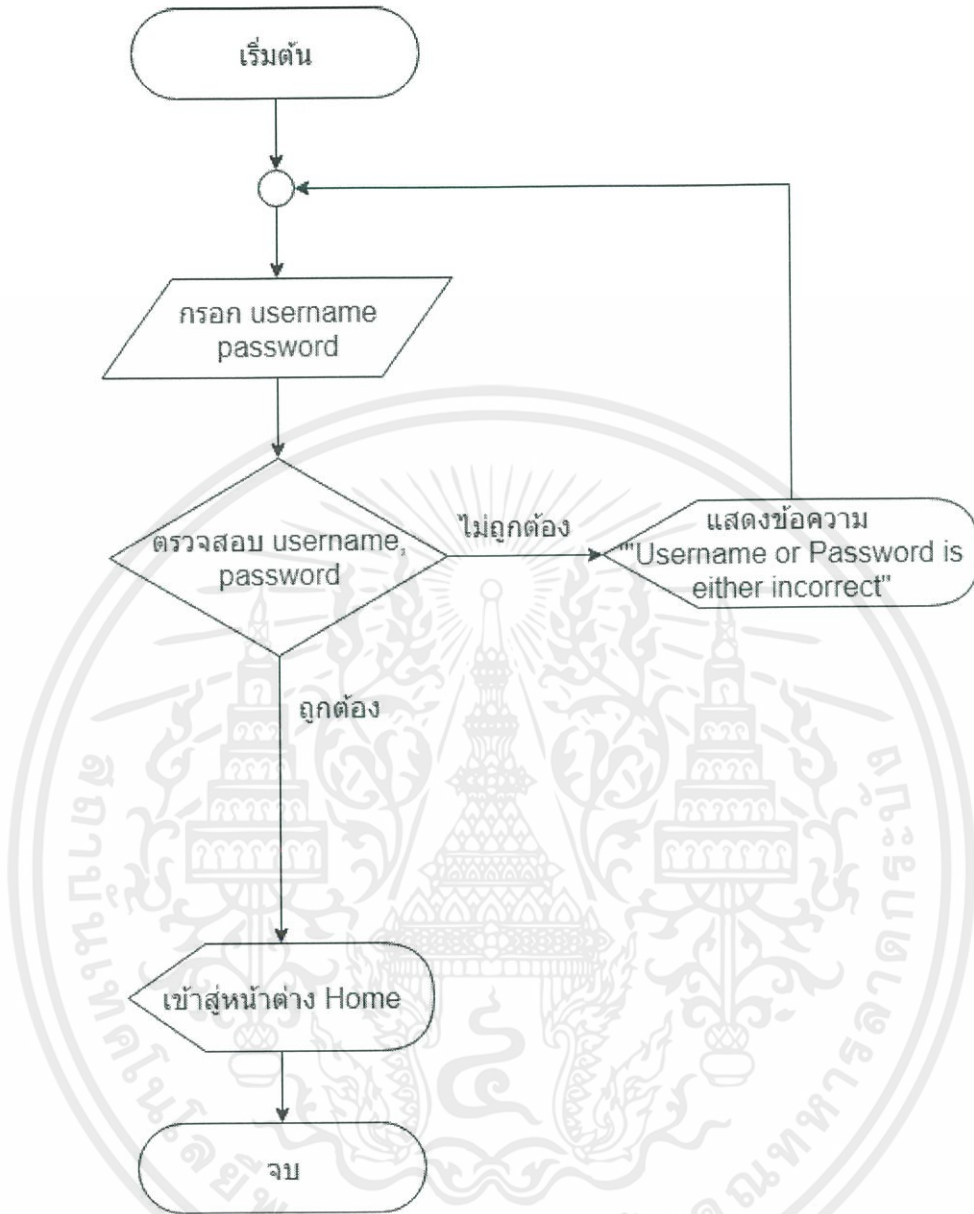
การออกแบบหน้าตาต่างแอปพลิเคชันสำหรับ Login สร้างเพื่อความปลอดภัย และควบคุมการใช้งานแอปพลิเคชันให้ใช้งานเฉพาะพนักงานของบริษัทหรือใช้งานเฉพาะพนักงานในแผนกในโครงการนี้จึงต้องมีหน้าตาต่างสำหรับ Login ซึ่งได้ออกแบบหน้าตาต่าง Login ดังรูปที่ 3.17 โดยมีการทำงานดังแผนผัง (flowchart) ดังแสดงในรูปที่ 3.18



(ก)

(ข)

รูปที่ 3.17 ออกแบบหน้าตาต่างสำหรับ Login (ก) Android และ (ข) iOS



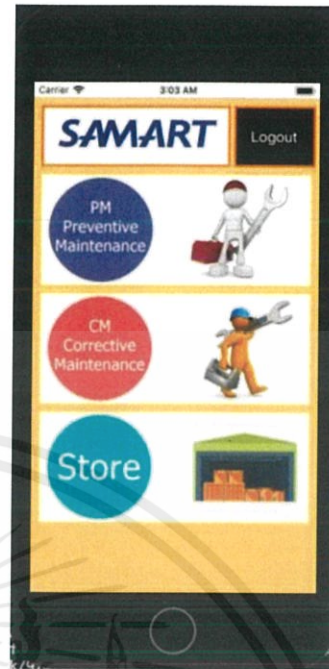
รูปที่ 3.18 แผนผังแสดงการทำงานของระบบ Login

3.1.3.2 ออกแบบหน้าต่างหลักของแอปพลิเคชัน

ออกแบบหน้าต่างหลักของแอปพลิเคชันมีไว้สำหรับเลือกใช้งานส่วนต่าง ๆ ประกอบไปด้วย 3 ส่วนคือ Preventive Maintenance, Corrective Maintenance และ Store หน้าต่างหลักของแอปพลิเคชันแสดงดังรูปที่ 3.19



(ก)



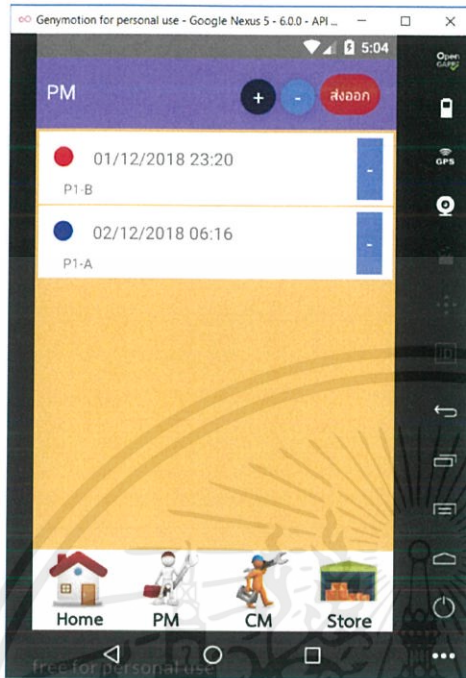
(ข)

รูปที่ 3.19 ออกแบบหน้าต่างหลักของแอปพลิเคชัน (ก) Android และ (ข) iOS

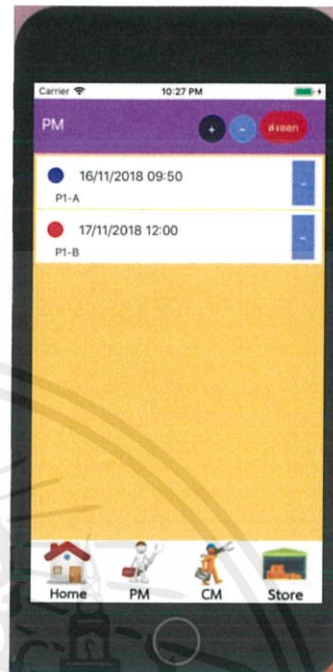
3.1.3.3 ออกแบบหน้าต่างสำหรับ Preventive Maintenance

Preventive Maintenance มีหลายหน้าเพื่อเป็นกระบวนการทำงานสำหรับ Preventive Maintenance ซึ่งหน้าแรกของ Preventive Maintenance คือรายการงานที่ต้องทำ (Schedule) แสดงดังรูปที่ 3.20

หน้าต่างสำหรับแสดงรายการงานที่ต้องทำประกอบไปด้วยปุ่มเพิ่มและลบรายการงานที่ต้องทำ ซึ่งสามารถทำได้เฉพาะพนักงานที่เป็นคนจัดสถานที่ เวลา ที่ต้องทำเท่านั้น โดยได้ป้องกันพนักงานอื่นด้วยการให้ใส่ Password ยืนยันอีกครั้งถึงจะสามารถเพิ่มและลบงานได้ดังรูปที่ 3.21 การลบรายการงานที่ต้องทำสามารถเลือกลบแค่บางรายการหรือลบรายการทั้งหมดได้ นอกจากปุ่มเพิ่มและลบงานแล้ว หน้าต่างแสดงรายการงานที่ต้องทำสามารถแสดงให้พนักงานสามารถรู้ได้ว่างานทำแล้วหรือไม่ โดยแสดงผลออกมาที่จุดด้านหน้ารายการงานของแต่ละงานถ้าเป็น “สีแดง” หมายความว่ารายการงานนั้นยังไม่มีการทำงานหรืองานนั้นยังไม่เสร็จสมบูรณ์ ถ้าเป็น “สีฟ้า” หมายความว่ารายการงานนั้นได้เสร็จสมบูรณ์ โดยเกิดจากพนักงานที่ปฏิบัติงานกดที่ปุ่ม “เสร็จงาน” ในรูปที่ 3.26

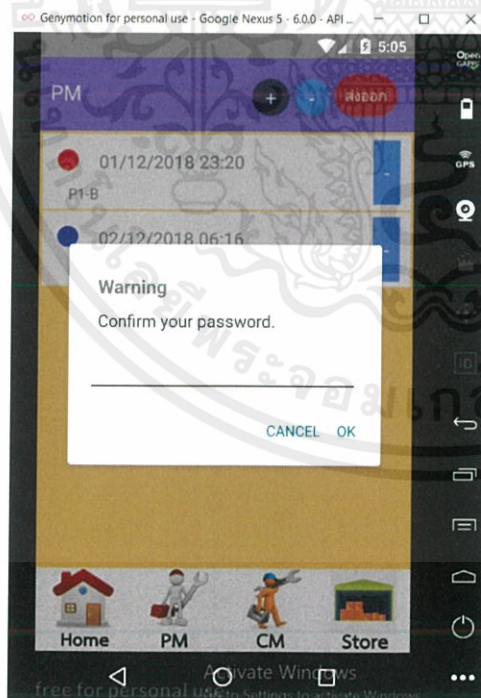


(ก)

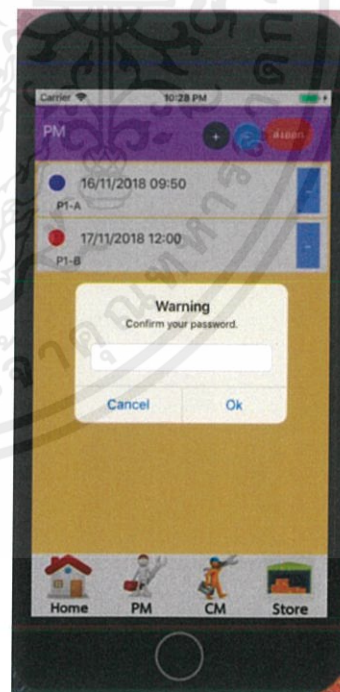


(ข)

รูปที่ 3.20 หน้าต่างรายการงานที่ต้องทำ (ก) Android และ (ข) iOS



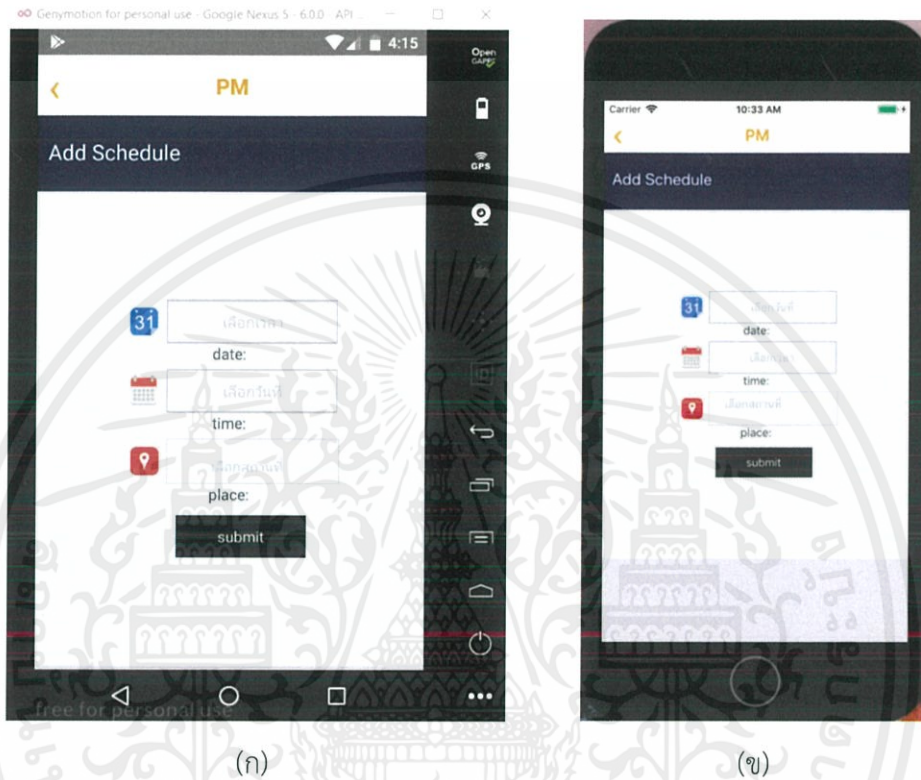
(ก)



(ข)

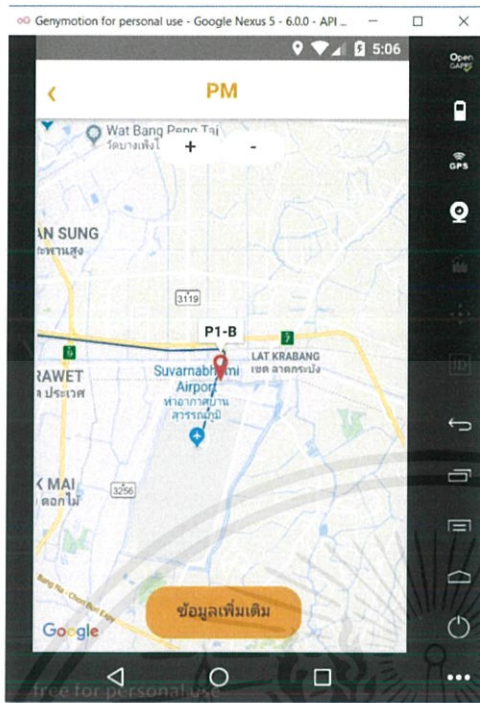
รูปที่ 3.21 ใส่ Password ยืนยันก่อนเพิ่มหรือลบงาน (ก) Android และ (ข) iOS

หลังจากกดปุ่มเพิ่มงานและใส่ Password ยืนยันตัวตนอีกครั้งว่าเป็นพนักงานที่สามารถเพิ่มรายการงานที่ต้องทำได้ จะสามารถเข้าไปเพิ่มงานได้ที่หน้าต่าง “Add Schedule” ซึ่งประกอบด้วย 3 อย่างที่ต้องระบุคือ วันเดือนปี เวลา และสถานที่ แสดงดังรูปที่ 3.22

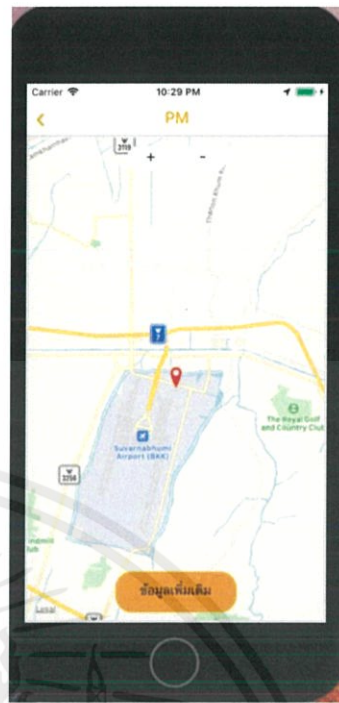


รูปที่ 3.22 หน้าต่าง Add Schedule (ก) Android และ (ข) iOS

จากรูปที่ 3.20 เป็นรูปรายการงานที่ต้องทำ เมื่อพนักงานต้องเข้าไปปฏิบัติงาน พนักงานกดรายการงานนั้น หลังจากนั้นจะเข้าสู่หน้าต่างสถานที่ที่พนักงานต้องไปปฏิบัติงาน ซึ่งจะป๊อปหุ้ดสีแสดงดังรูปที่ 3.23 ถ้าหากพนักงานไม่รู้เส้นทางไปยังสถานที่นั้น ๆ พนักงานสามารถกดที่หุ้ดสีแดง แอปพลิเคชันจะแสดงให้เลือกเปิดไปที่ “Google Maps” ดังรูปที่ 3.24 และรูปที่ 3.25 เพื่อที่จะทำให้นาทงไปยังสถานที่นั้นได้ หรือถ้าหากพนักงานรู้เส้นทางไปยังสถานที่ที่ต้องไปปฏิบัติงานแล้ว ก็ไม่จำเป็นต้องเปิด “Google Maps”



(ก)

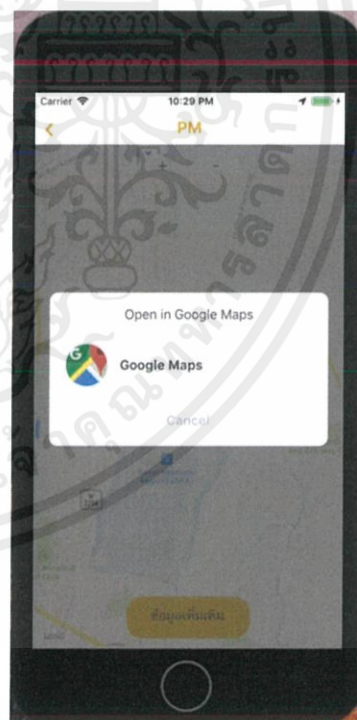


(ข)

รูปที่ 3.23 หน้าต่างการปิดหมดสถานที่ (ก) Android และ (ข) iOS



(ก)

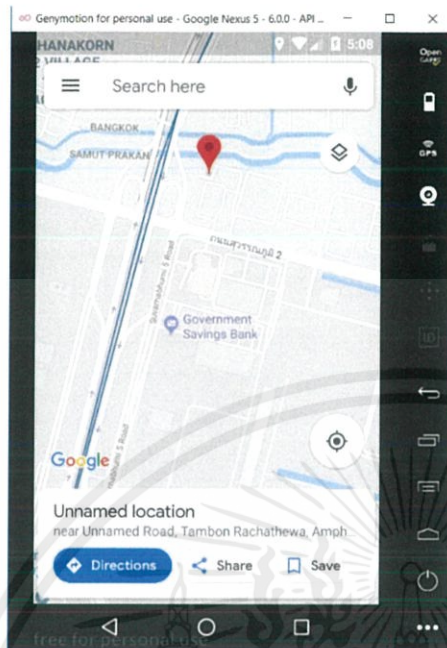


(ข)

รูปที่ 3.24 เลือกเปิดไปที่ Google Maps ของ Preventive Maintenance

(ก) Android และ (ข) iOS

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูปที่ 3.25 หน้าต่าง Google Maps ของ Preventive Maintenance

(ก) Android และ (ข) iOS

จากรูปที่ 3.23 จะมีปุ่ม “ข้อมูลเพิ่มเติม” เพื่อกดดูอุปกรณ์ที่มีอยู่ตามสถานที่นั้น ๆ ซึ่งเป็นอุปกรณ์ต้องทำ Preventive Maintenance ดังรูปที่ 3.26 ซึ่งเมื่อปฏิบัติงานเสร็จเรียบร้อยแล้ว ให้กดที่ปุ่ม “เสร็จงาน” เพื่อไปแสดงข้อมูลว่าได้ทำ Preventive Maintenance เสร็จแล้วในหน้าต่างรูปที่ 3.20



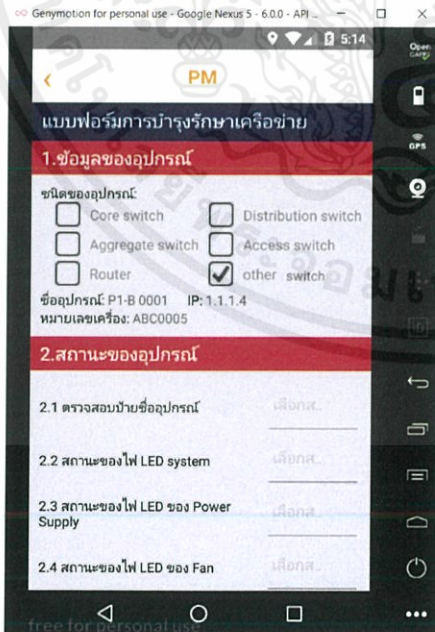
(ก)



(ข)

รูปที่ 3.26 หน้าต่างข้อมูลเพิ่มเติมของ Preventive Maintenance (ก) Android และ (ข) iOS

จากรูปที่ 3.26 จะมีปุ่มด้านข้างชื่อของอุปกรณ์เพื่อเข้าไปกรอกที่หน้าต่าง “แบบฟอร์มการบำรุงรักษาเครือข่าย” ดังรูปที่ 3.27 ซึ่งมีชื่ออุปกรณ์ หมายเลข IP และหมายเลขเครื่อง



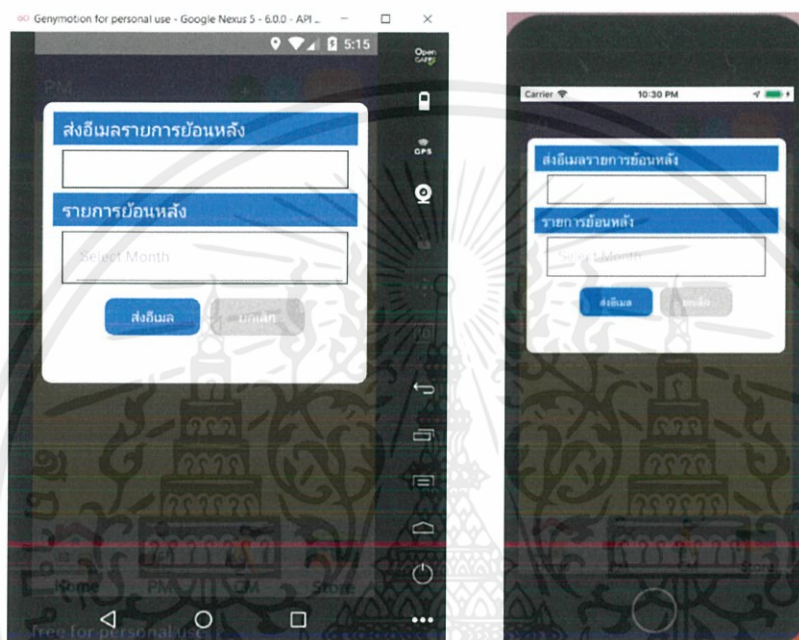
(ก)



(ข)

รูปที่ 3.27 หน้าต่างแบบฟอร์มการบำรุงรักษาเครือข่าย (ก) Android และ (ข) iOS

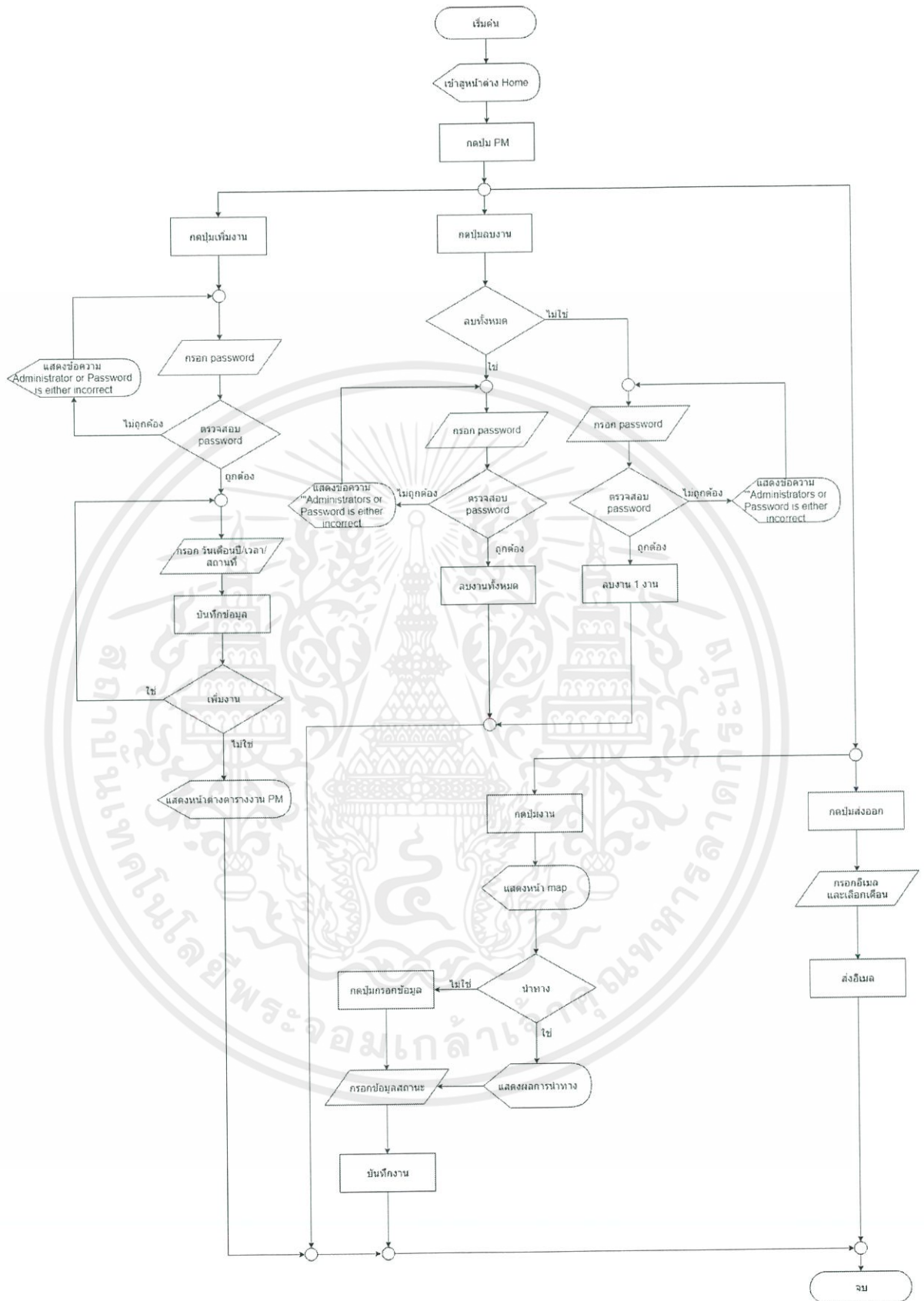
นอกจากนี้สามารถกดที่ปุ่มส่งออกในหน้าต่างรูปที่ 3.20 เพื่อทำการส่งออกข้อมูลแบบฟอร์มการบำรุงรักษาเครือข่ายและสถานะของการทำงานในแต่ละงาน โดยส่งออกเป็นไฟล์ Excel ผ่านทางอีเมลแสดงในรูปที่ 3.28 โดยระบบสำหรับ Preventive Maintenance มีการทำงานดังแผนผัง ดังแสดงในรูปที่ 3.29



(ก)

(ข)

รูปที่ 3.28 หน้าต่างการส่งข้อมูลสถานะของอุปกรณ์ไปยังอีเมล (ก) Android และ (ข) iOS



รูปที่ 3.29 แผนผังแสดงการทำงานของระบบ Preventive Maintenance

3.1.3.4 ออกแบบหน้าตาสำหรับ Corrective Maintenance

การสร้างหน้าตาสำหรับงาน Corrective Maintenance มีหลายหน้าตาต่าง ซึ่งหน้าแรกคือหน้าตาหลักจะมีตัวเลือกทั้งหมด 4 ตัวเลือกในการทำงานคือ การเตรียมเข้าปฏิบัติงาน เบื้องต้น การค้นหาเส้นทาง การแจ้งเปลี่ยนอุปกรณ์ และรายการปัญหาที่เกิดขึ้นแล้วแสดงดังรูปที่ 3.30



(ก)

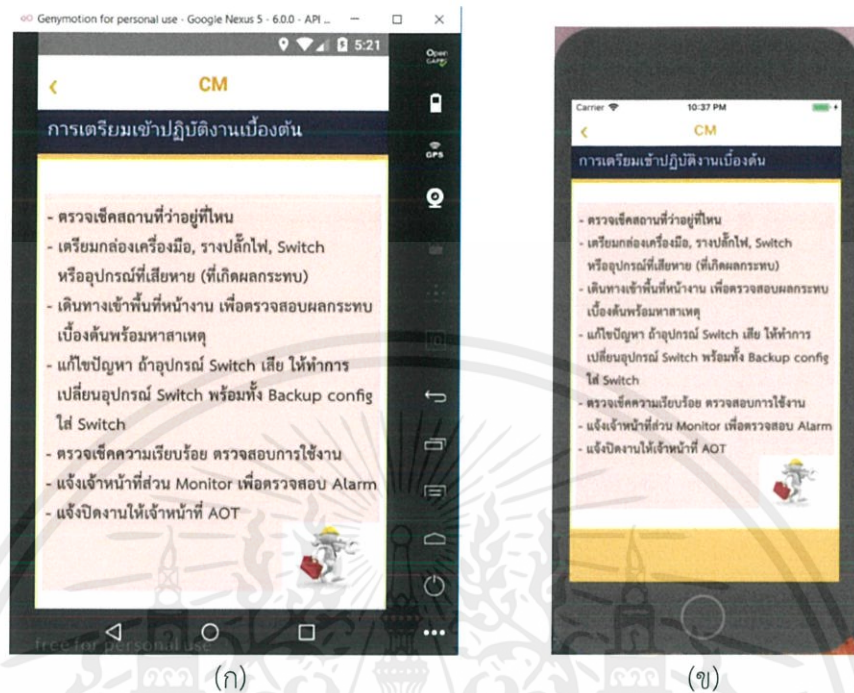


(ข)

รูปที่ 3.30 หน้าตาหลักของ Corrective Maintenance (ก) Android และ (ข) iOS

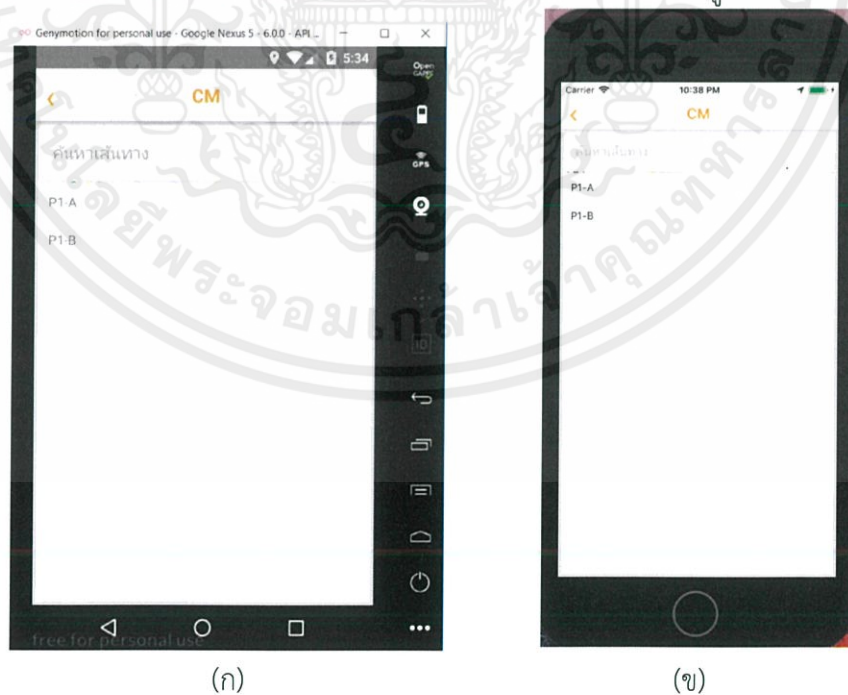
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าตาการเตรียมเข้าปฏิบัติงานเบื้องต้น แสดงถึงขั้นตอนเบื้องต้นในการปฏิบัติงาน แสดงดังรูปที่ 3.31



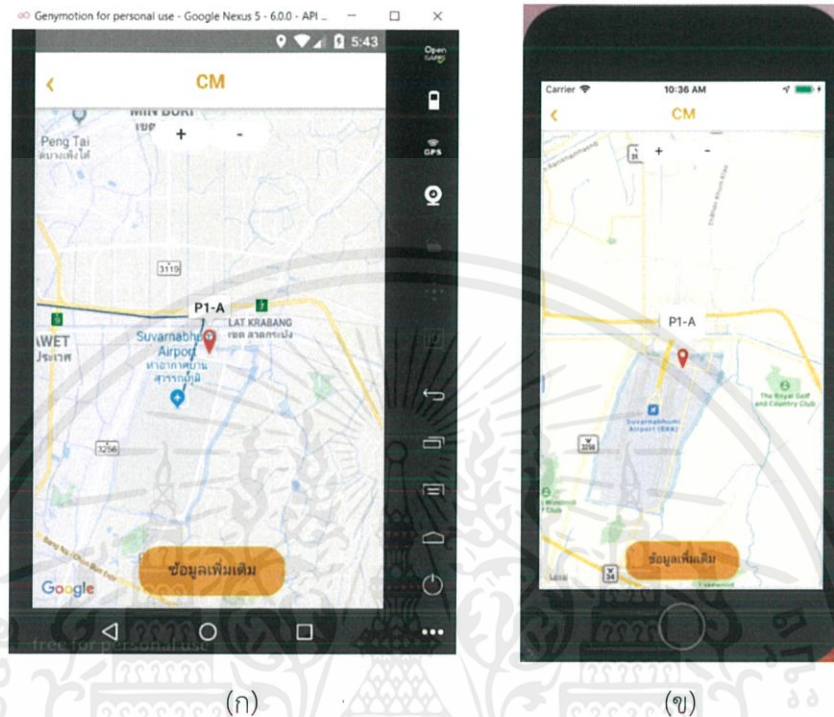
รูปที่ 3.31 หน้าตาการเตรียมตัวเข้าปฏิบัติงานเบื้องต้น (ก) Android และ (ข) iOS

จากรูปที่ 3.30 เมื่อพนักงานเลือก “เส้นทาง” เพื่อเข้าไปค้นหาสถานที่ที่ต้องไปปฏิบัติงาน Corrective Maintenance จะมีหน้าตาการค้นหาเส้นทางดังรูปที่ 3.32



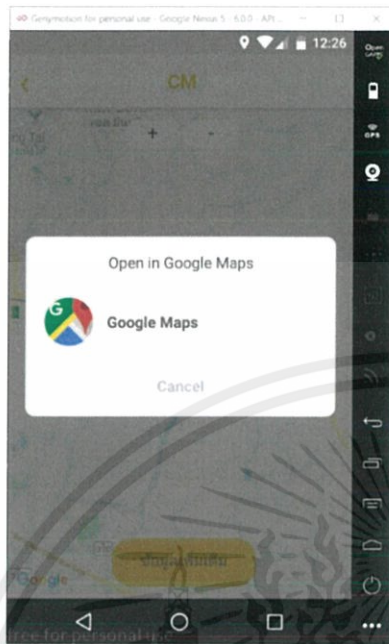
รูปที่ 3.32 หน้าตาการค้นหาเส้นทาง (ก) Android และ (ข) iOS

เมื่อกดค้นหาเส้นทางจากหน้าต่างรูปที่ 3.32 แล้ว จะสามารถไปยังหน้า
สถานที่นั้นๆ ที่ปักหมุดไว้บนแผนที่ ดังรูปที่ 3.33

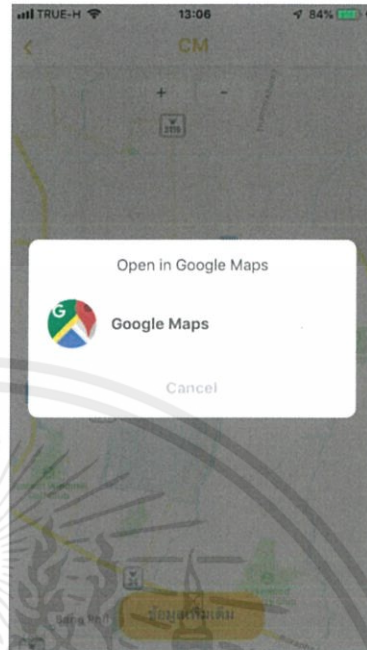


รูปที่ 3.33 หน้าต่างการปักหมุดสถานที่ (ก) Android และ (ข) iOS

ถ้าหากพนักงานไม่รู้เส้นทางไปยังสถานที่นั้น ๆ พนักงานสามารถกดที่หมุดสีแดง แอปพลิเคชันจะแสดงให้เลือกเปิดไปที่ “Google Maps” ดังรูปที่ 3.33 และรูปที่ 3.34 เพื่อที่จะ
ทำให้นำทางไปยังสถานที่นั้นได้ หรือถ้าหากพนักงานรู้เส้นทางไปยังสถานที่ที่ต้องไปปฏิบัติงาน ไม่
จำเป็นต้องเปิด “Google Maps” เหมือนกันกับการนำเส้นทางของ Preventive Maintenance



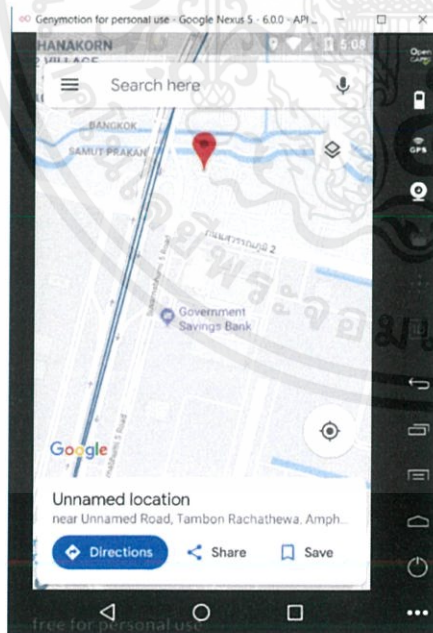
(ก)



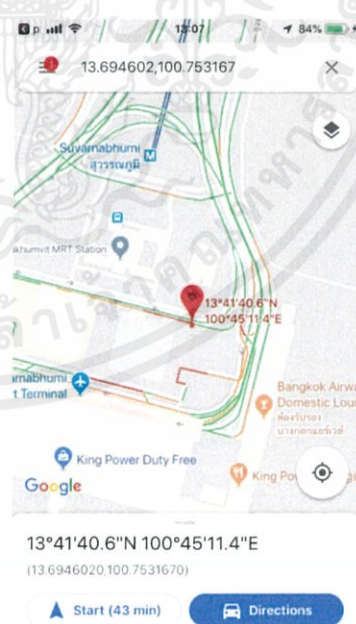
(ข)

รูปที่ 3.34 เลือกเปิดไปที่ Google Maps ของ Corrective Maintenance

(ก) Android และ (ข) iOS



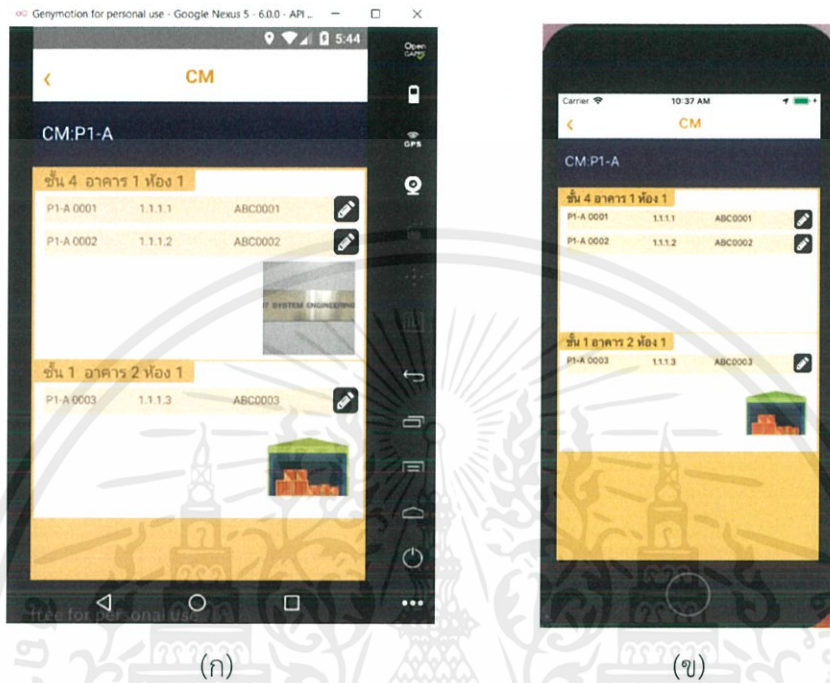
(ก)



(ข)

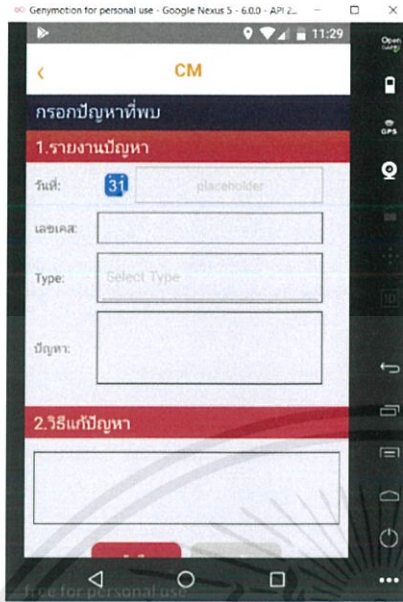
รูปที่ 3.35 หน้าต่าง Google Maps ของ Corrective Maintenance (ก) Android และ (ข) iOS

จากรูปที่ 3.33 จะมีปุ่ม “ข้อมูลเพิ่มเติม” เพื่อกดดูอุปกรณ์ที่มีอยู่ตามสถานที่นั้น ซึ่งงานที่เป็นงาน Corrective Maintenance จะเลือกทำเฉพาะอุปกรณ์ที่มีปัญหา โดยหน้าข้อมูลเพิ่มเติมแสดงดังรูปที่ 3.36

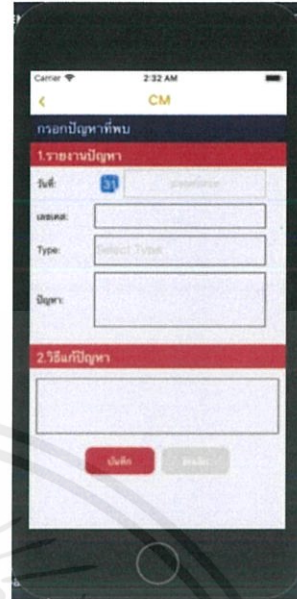


รูปที่ 3.36 หน้าต่างข้อมูลเพิ่มเติมของ Corrective Maintenance (ก) Android และ (ข) iOS

จากรูปที่ 3.36 จะมีปุ่มด้านข้างชื่ออุปกรณ์เพื่อเข้าไปที่หน้าต่าง “กรอกปัญหาที่พบ” ดังรูปที่ 3.37 ซึ่งเมื่อพนักงานเจอปัญหาของอุปกรณ์ตัวใด กดปุ่มของอุปกรณ์ตัวนั้นเพื่อเข้าไปกรอกรายงานปัญหาและวิธีแก้ปัญหของตัวอุปกรณ์นั้น ซึ่งข้อมูลที่กรอกไปจะไปแสดงผลไว้ในหน้าต่าง “ปัญหาที่เกิดขึ้นแล้ว” ในรูปที่ 3.40



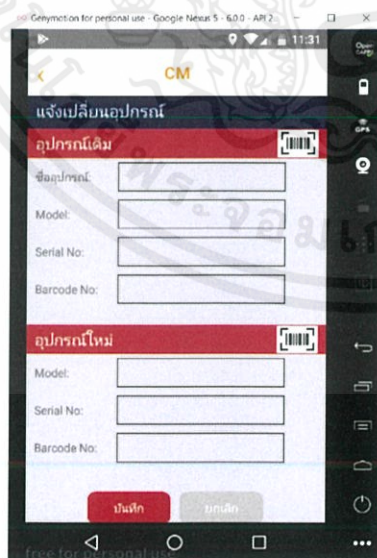
(ก)



(ข)

รูปที่ 3.37 หน้าต่างกรอกปัญหาที่พบ (ก) Android และ (ข) iOS

จากรูปที่ 3.30 ตัวเลือก “แจ้งเปลี่ยนอุปกรณ์” เมื่อเจอปัญหาอุปกรณ์ที่ต้องเปลี่ยนอุปกรณ์ พนักงานจะนำอุปกรณ์ออกจากคลังพัสดุและนำอุปกรณ์มาเปลี่ยนแทนที่อุปกรณ์เดิมที่มีอยู่ในสถานะนั้น โดยพนักงานต้องแจ้งเปลี่ยนอุปกรณ์เพื่อให้ข้อมูลรายละเอียดอุปกรณ์แต่ละสถานที่หรือสถานที่ที่เปลี่ยนอัปเดตใหม่ โดยหน้าต่างแจ้งเปลี่ยนอุปกรณ์แสดงดังรูปที่ 3.38



(ก)



(ข)

รูปที่ 3.38 หน้าต่างการแจ้งเปลี่ยนอุปกรณ์ (ก) Android และ (ข) iOS

จากรูปที่ 3.38 การแจ้งเปลี่ยนอุปกรณ์สามารถเลือกปุ่มตัวสแกนคิวอาร์โค้ดเพื่อสแกนคิวอาร์โค้ดที่ตัวอุปกรณ์ได้โดยที่ไม่ต้องกรอกข้อมูลเอง ซึ่งข้อมูลของตัวอุปกรณ์ที่สแกนจะแสดงชื่ออุปกรณ์, โมเดล, หมายเลขเครื่องและเลขครุภัณฑ์ หน้าต่างตัวอ่านบาร์โค้ดดังรูปที่ 3.39



รูปที่ 3.39 หน้าต่างตัวอ่านคิวอาร์โค้ด (ก) Android และ (ข) iOS

จากรูปที่ 3.30 ตัวเลือก “ปัญหาที่เกิดขึ้นแล้ว” เป็นหน้าต่างที่ให้พนักงานสามารถเข้าไปดูปัญหาที่เคยเกิดขึ้นพร้อมทั้งแสดงรายละเอียด ได้แก่ วันที่เกิดปัญหา เลขเคส ชนิดหรืออุปกรณ์ที่เกิด ปัญหา วิธีแก้ และชื่ออุปกรณ์ที่เกิดปัญหา ซึ่งแสดงดังรูปที่ 3.40



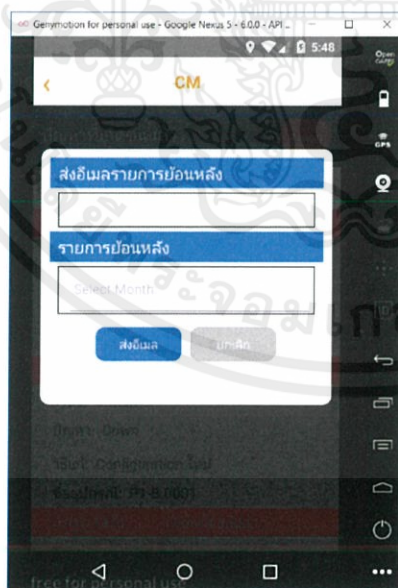
(ก)



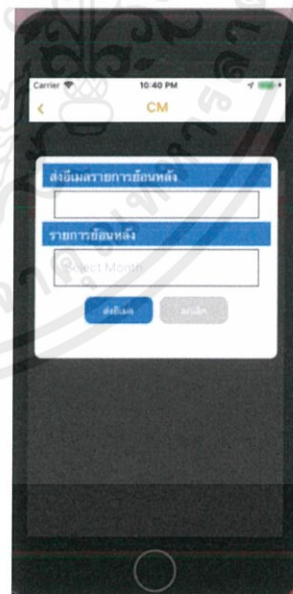
(ข)

รูปที่ 3.40 หน้าต่างปัญหาที่เกิดขึ้นแล้ว (ก) Android และ (ข) iOS

นอกจากนี้สามารถกดที่ปุ่มส่งออกในหน้าต่างรูปที่ 3.30 เพื่อทำการส่งออกข้อมูลปัญหาที่เคยเกิดขึ้น โดยส่งออกเป็นไฟล์ Excel ผ่านทางอีเมลแสดงในรูปที่ 3.41 โดยระบบสำหรับ Corrective Maintenance มีการทำงานดังแผนผัง ดังแสดงในรูปที่ 3.42

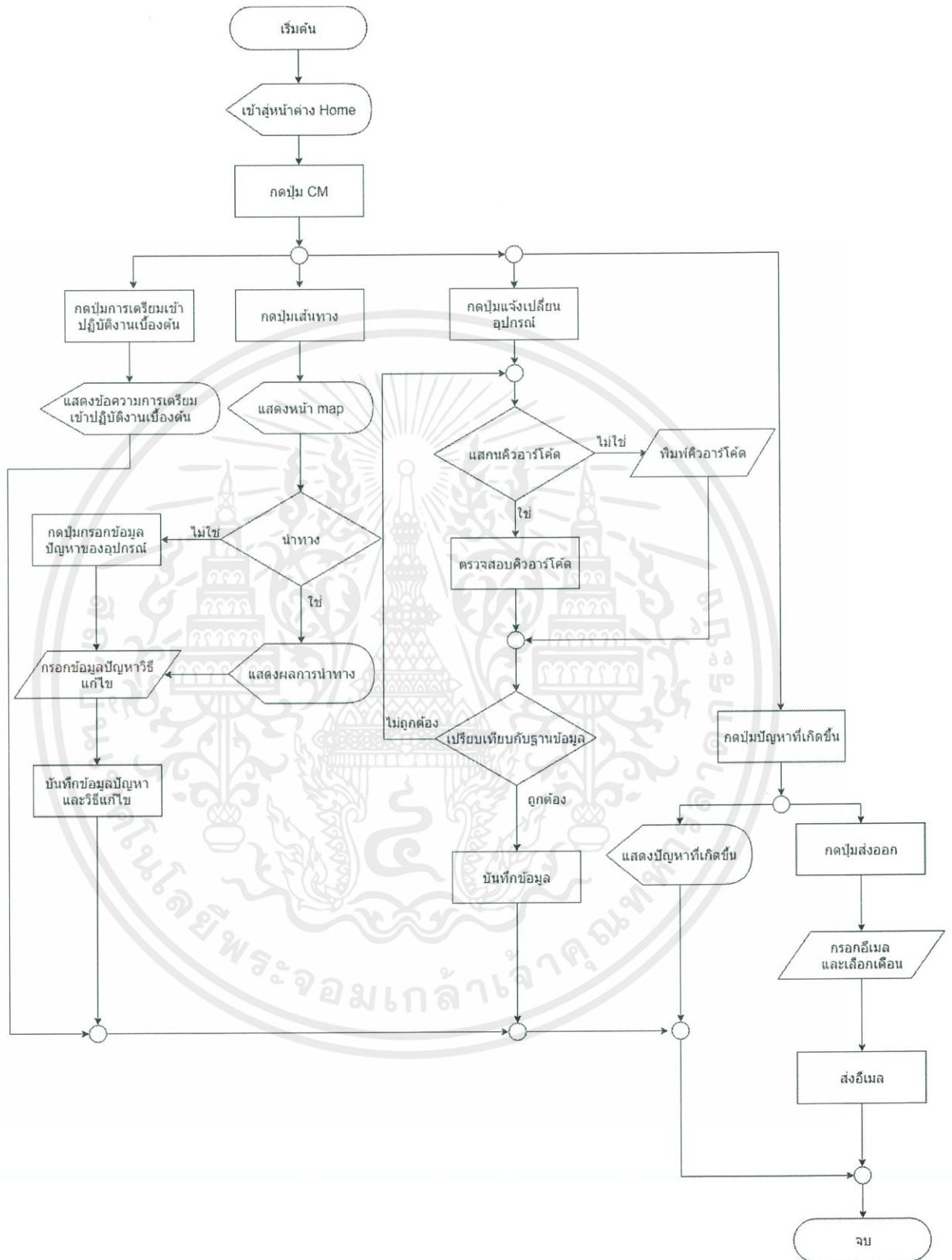


(ก)



(ข)

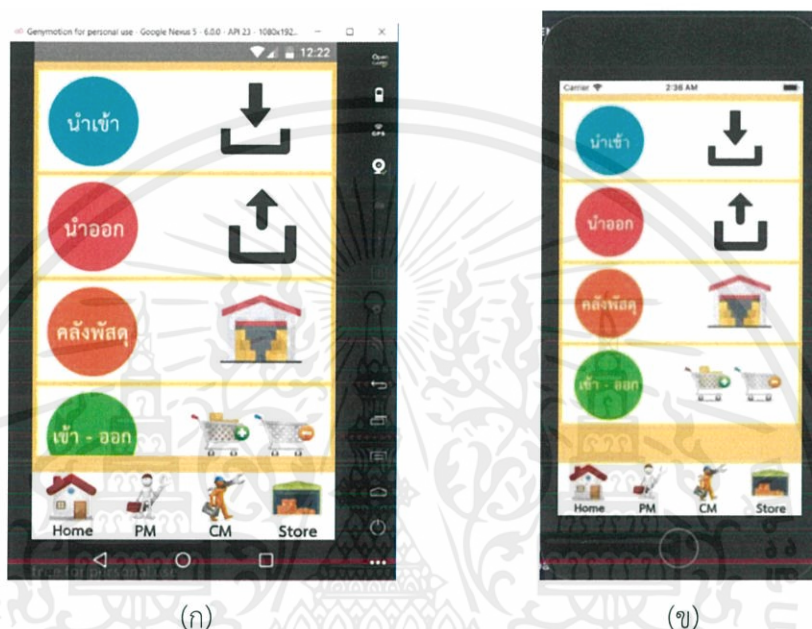
รูปที่ 3.41 หน้าต่างการส่งข้อมูลการปัญหาผ่านอีเมล (ก) Android และ (ข) iOS



รูปที่ 3.42 แผนผังแสดงการทำงานของระบบ Corrective Maintenance

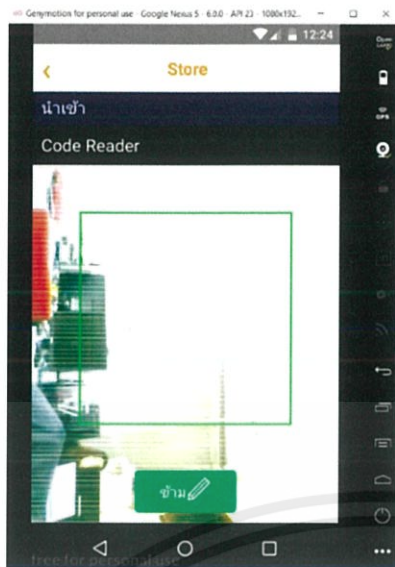
3.1.3.4 ออกแบบต่างสำหรับคลังพัสดุ

การสร้างหน้าต่างของคลังพัสดุ เพื่ออำนวยความสะดวกในการบริหารจัดการคลังพัสดุ โดยหน้าแรกของคลังพัสดุ คือหน้าต่างหลักของคลังพัสดุ ซึ่งมีตัวเลือกให้เข้าใช้งานทั้งหมด 4 ตัวเลือกคือ นำเข้า นำออก คลังพัสดุ และเข้า-ออก แสดงดังรูปที่ 3.43

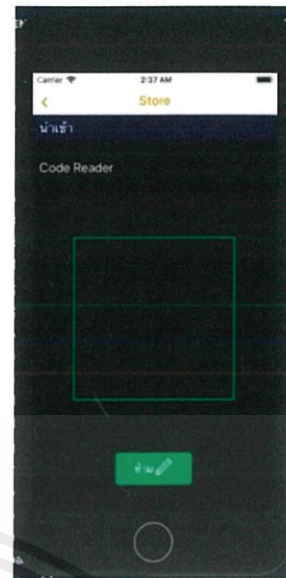


รูปที่ 3.43 หน้าต่างหลักของคลังพัสดุ (ก) Android และ (ข) iOS

จากรูปที่ 3.43 เมื่อต้องการนำอุปกรณ์เข้าคลังพัสดุ สามารถเลือกเมนู “นำเข้า” เพื่อไปที่หน้าต่างตัวอ่านคิวอาร์โค้ดเพื่อสแกนอุปกรณ์ที่ต้องการนำเข้าคลังพัสดุ ดังรูปที่ 3.44 และสามารถกดข้ามหน้าต่างตัวอ่านคิวอาร์โค้ดนำเข้าได้เมื่อต้องการกรอกข้อมูลเองทั้งหมด โดยแสดงหน้ากรอกข้อมูลนำเข้าในรูปที่ 3.45

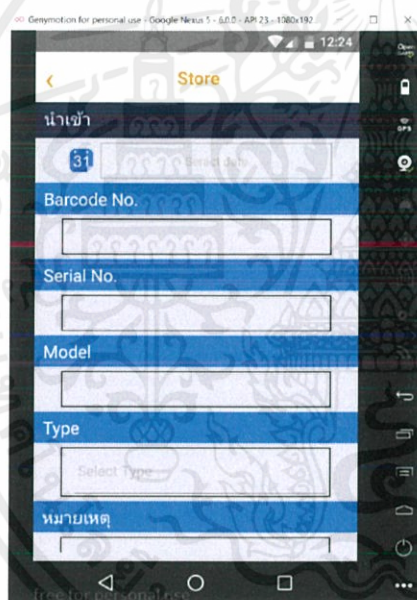


(ก)

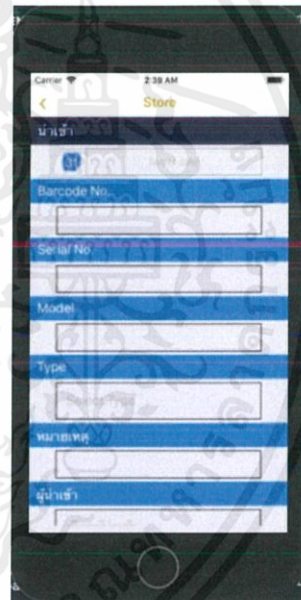


(ข)

รูปที่ 3.44 หน้าต่างตัวอ่านคิวอาร์โค้ดนำเข้าคลังพัสดุ (ก) Android และ (ข) iOS



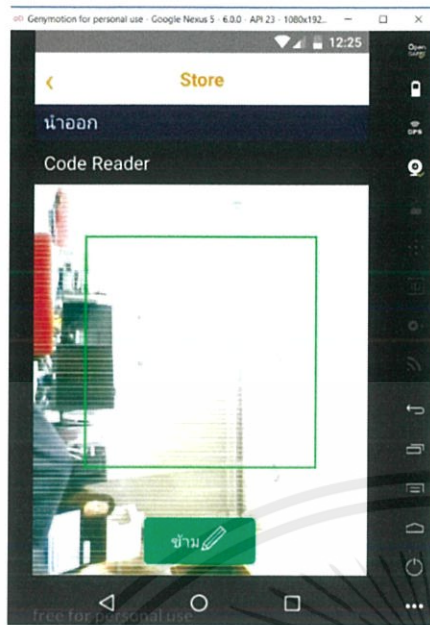
(ก)



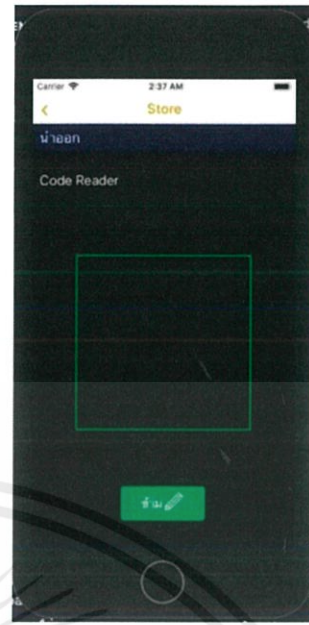
(ข)

รูปที่ 3.45 หน้าต่างกรอกข้อมูลอุปกรณ์นำเข้าของคลังพัสดุ (ก) Android และ (ข) iOS

จากรูปที่ 3.43 เมื่อต้องการนำอุปกรณ์ออกจากคลังพัสดุ สามารถเลือกเมนู “นำออก” เพื่อไปที่หน้าต่างตัวอ่านคิวอาร์โค้ดเพื่อสแกนอุปกรณ์ที่ต้องการนำออกจากคลังพัสดุ ดังรูปที่ 3.46 เมื่อสแกนคิวอาร์โค้ดนำออกจะทำให้ข้อมูลในหน้าต่างกรอกข้อมูลอุปกรณ์นำออกในหน้าที่ 3.47 แสดงบางรายการโดยที่ไม่ต้องกรอกข้อมูลเองคือ Barcode No., Serial No., Model และ Type ซึ่งสามารถกดข้ามหน้าต่างตัวอ่านบาร์โค้ดนำออกได้เมื่อต้องการกรอกข้อมูลเองทั้งหมด

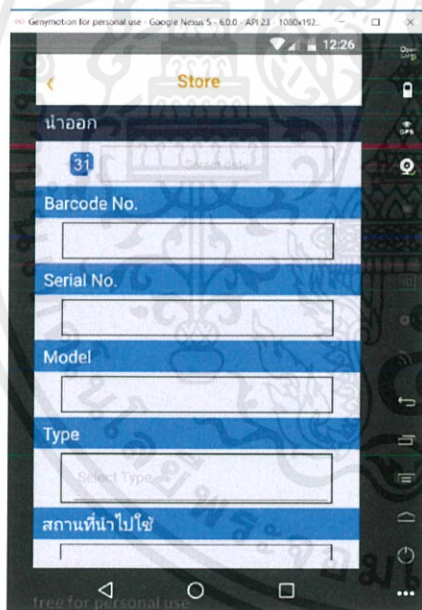


(ก)

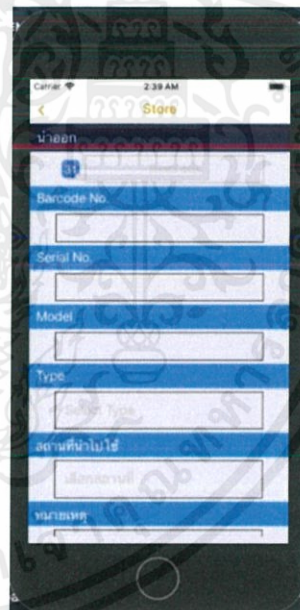


(ข)

รูปที่ 3.46 หน้าต่างตัวอ่านคิวอาร์โค้ดนำออกจากคลังพัสดุ (ก) Android และ (ข) iOS



(ก)

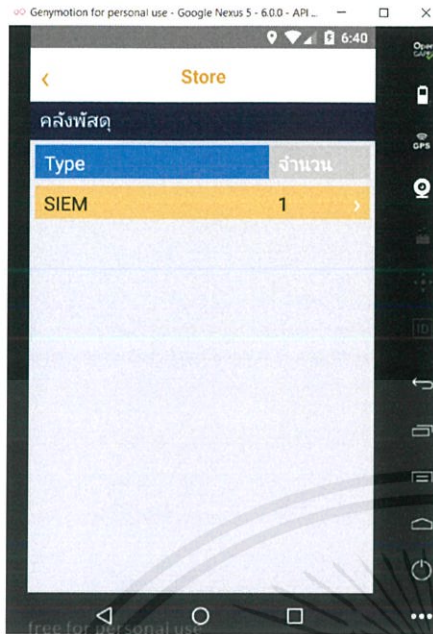


(ข)

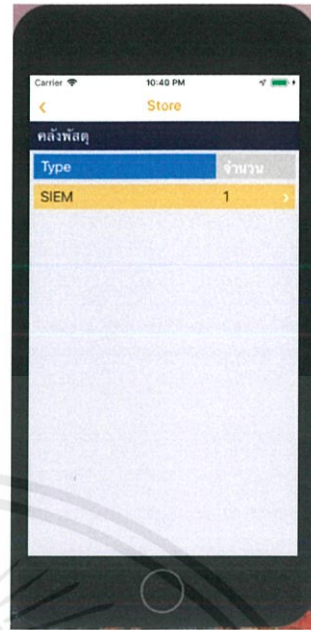
รูปที่ 3.47 หน้าต่างกรอกข้อมูลอุปกรณ์นำออกจากคลังพัสดุ (ก) Android และ (ข) iOS

จากรูปที่ 3.43 เมื่อต้องการดูข้อมูลอุปกรณ์ที่อยู่ในคลังพัสดุสามารถเลือกเมนู “คลังพัสดุ” โดยจะแสดงข้อมูลประเภทของอุปกรณ์และบอกจำนวนของแต่ละประเภทดังรูปที่ 3.48 และเมื่อต้องการดูว่าแต่ละประเภทมีรายละเอียดของอุปกรณ์ตัวไหนอยู่บ้างให้กดเข้าไปในประเภท (Type) ของอุปกรณ์นั้น เช่น เมื่อต้องการรู้รายละเอียดว่า Switch ทั้งหมดมีข้อมูลที่เป็น Model อะไรและ Serial No. อะไร ให้กดประเภท Switch จะได้ข้อมูลแสดงดังรูปที่ 3.49

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

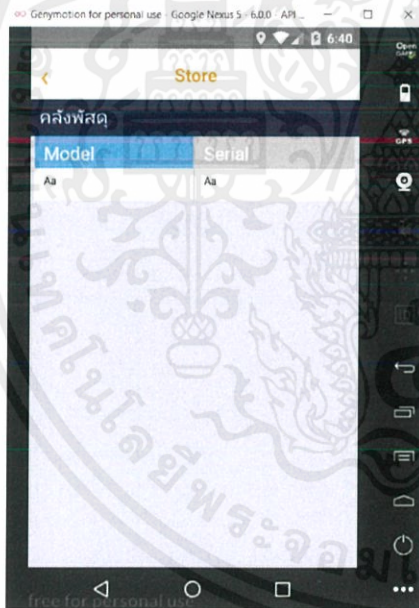


(ก)

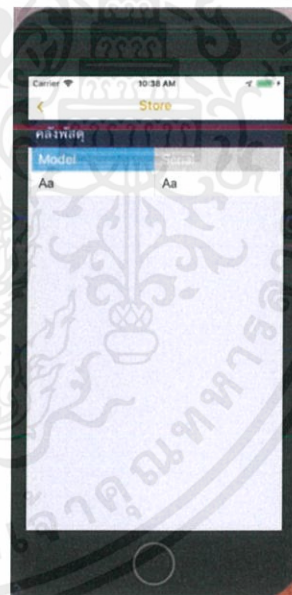


(ข)

รูปที่ 3.48 หน้าต่างคลังพัสดุแยกตามประเภท (ก) Android และ (ข) iOS



(ก)



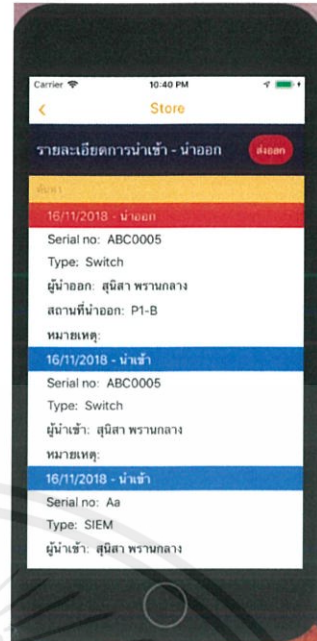
(ข)

รูปที่ 3.49 หน้าต่างรายละเอียดของแต่ละประเภท (ก) Android และ (ข) iOS

จากรูปที่ 3.43 เมื่อต้องการดูรายละเอียดการนำเข้าและนำออกของอุปกรณ์ในคลังพัสดุ สามารถเลือกที่เมนู “เข้า – ออก” ดังรูปที่ 3.50 ซึ่งแสดงรายละเอียดการนำเข้าและออกทั้งหมดของ Store จะแสดงข้อมูลได้แก่ วันที่, สถานะนำเข้า (สีฟ้า) หรือนำออก (สีแดง), Serial No., Type, ชื่อผู้นำเข้า หรือนำออกและหมายเหตุ



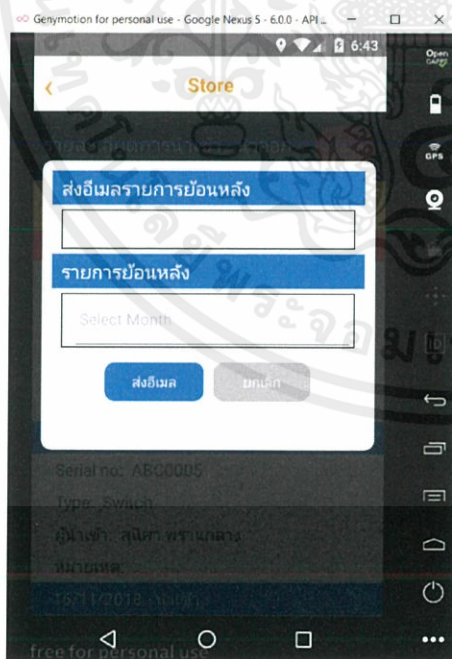
(ก)



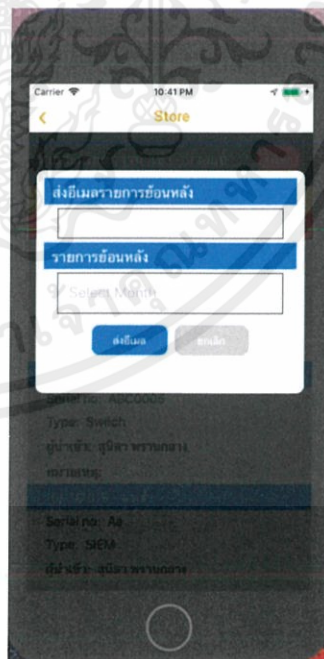
(ข)

รูปที่ 3.50 หน้าต่างรายละเอียดการนำเข้าและนำออกจากคลังพัสดุ (ก) Android และ (ข) iOS

นอกจากนี้สามารถกดที่ปุ่มส่งออกในหน้าต่างรูปที่ 3.50 เพื่อทำการส่งออกข้อมูลการนำเข้า – ออกของอุปกรณ์ในแต่ละเดือน โดยส่งออกเป็นไฟล์ Excel ผ่านทางอีเมลแสดงในรูปที่ 3.51 โดยระบบสำหรับคลังพัสดุและอุปกรณ์ มีการทำงานดังแผนผัง ดังแสดงในรูปที่ 3.52

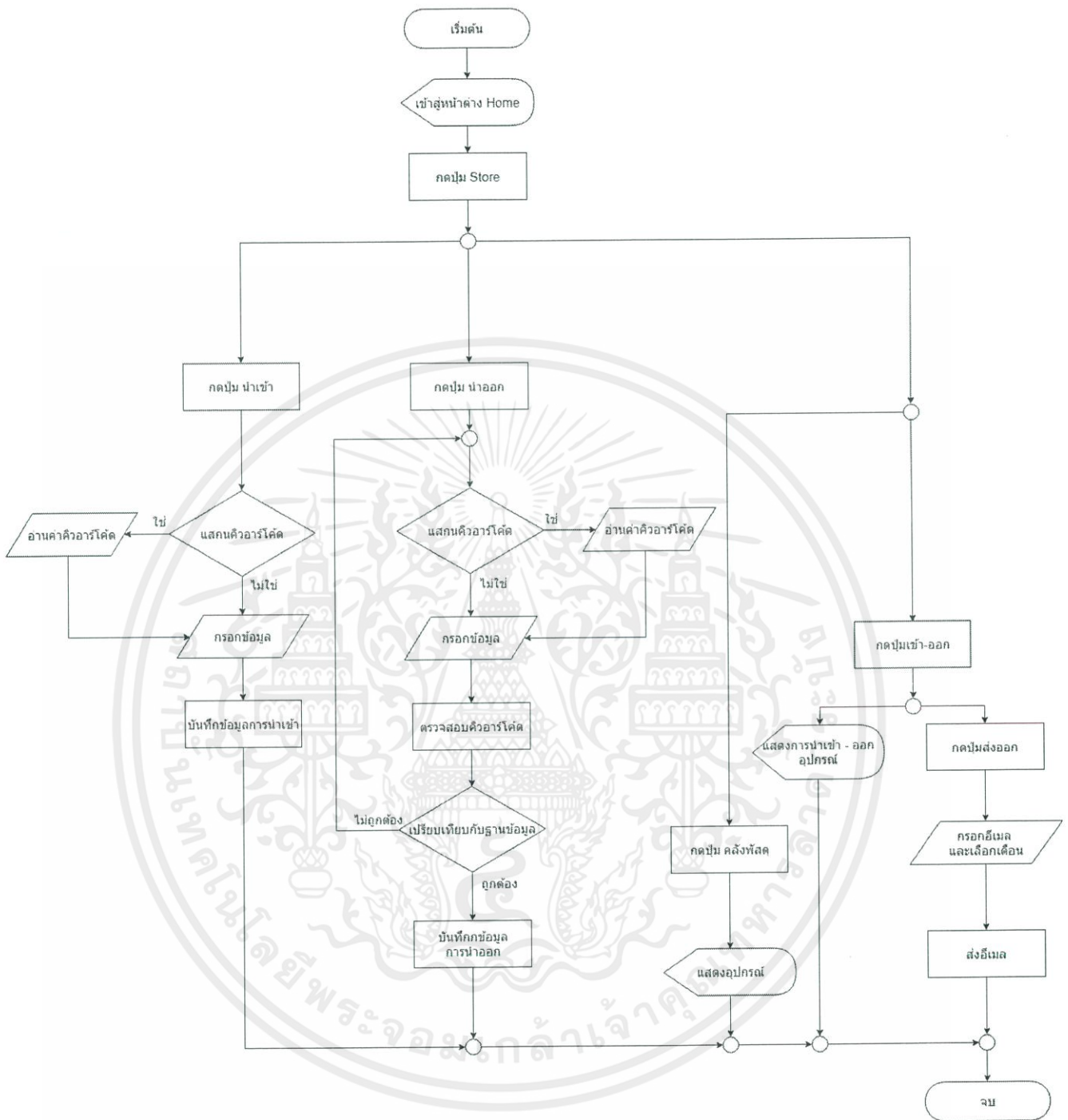


(ก)



(ข)

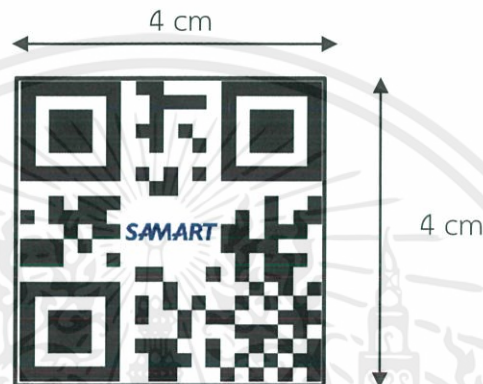
รูปที่ 3.51 หน้าต่างการส่งข้อมูลนำเข้า – ออกไปยังอีเมล (ก) Android และ (ข) iOS



รูปที่ 3.52 แผนผังแสดงการทำงานของระบบคลังพัสดุ

3.1.4 การออกแบบเลขครุภัณฑ์คิวอาร์โค้ดของอุปกรณ์ภายในห้องจัดเก็บพัสดุ

ในระบบห้องจัดเก็บพัสดุ ผู้จัดทำได้จัดทำเลขครุภัณฑ์ให้กับแผนกสำหรับอุปกรณ์ชนิดต่างๆที่อยู่ภายในห้องจัดเก็บพัสดุ รวมถึงอุปกรณ์ที่ถูกติดตั้งตามสถานที่ต่างๆภายใต้การกำกับดูแลของแผนก โดยทำการปรับเปลี่ยนรูปแบบตัวอักษรของเลขครุภัณฑ์ให้มาอยู่ในรูปแบบของคิวอาร์โค้ด และใช้ในการติดลงบนอุปกรณ์ โดยจัดทำขนาดความสูง 4 เซนติเมตร และความกว้าง 4 เซนติเมตร ที่แสดงในรูปที่ 3.53



รูปที่ 3.53 การออกแบบคิวอาร์โค้ดเลขครุภัณฑ์ของอุปกรณ์

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

สำหรับโครงการนี้ได้ใช้โทรศัพท์เคลื่อนที่ยี่ห้อ OPPO ในการทดสอบแอปพลิเคชัน ดังแสดงในรูปที่ 3.54 เป็นระบบปฏิบัติการแอนดรอยด์ OPPO A57 มีขนาด 149.1 มิลลิเมตร x 72.9 มิลลิเมตร x 7.65 มิลลิเมตร และความกว้างของจอแสดงผล 5.2 นิ้ว (แนวทแยง)



รูปที่ 3.54 โทรศัพท์เคลื่อนที่ยี่ห้อ OPPO

3.2.2 โทรศัพท์เคลื่อนที่ระบบปฏิบัติการไอโอเอส

สำหรับโครงการนี้ได้ใช้โทรศัพท์เคลื่อนที่ยี่ห้อ Apple ในการทดสอบแอปพลิเคชัน ดังแสดงในรูปที่ 3.55 เป็นระบบปฏิบัติการ iOS Apple SE มีขนาด 123.8 มิลลิเมตร x 58.6 มิลลิเมตร x 7.6 มิลลิเมตร และความกว้างของจอแสดงผล 4 นิ้ว (แนวทแยง)



รูปที่ 3.55 โทรศัพท์เคลื่อนที่ยี่ห้อ Apple

3.3 การจับเก็บผลการทดลอง

นำแอปพลิเคชันจริงมาติดตั้งลงบนโทรศัพท์มือถือจริง เพื่อทดสอบการใช้งานในสถานการณ์จริง โดยทำการจับเก็บผลการทดลองด้วยการบันทึกผล (Capture Screen) ในสิ่งที่เกิดขึ้นในหน้าต่าง ๆ ของส่วนต่อประสานกราฟิกกับผู้ใช้

บทที่ 4

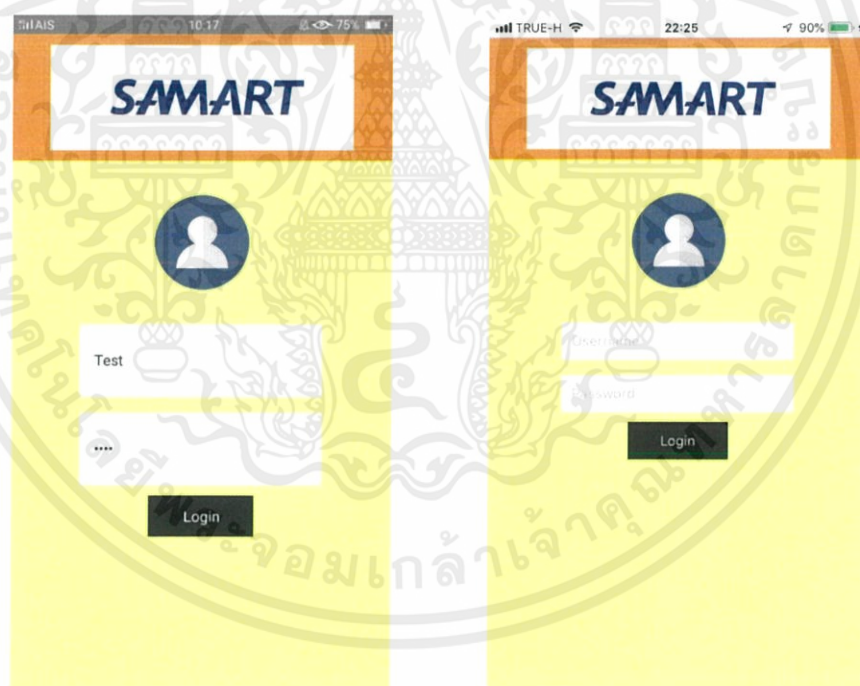
ผลการดำเนินงาน

ในบทนี้จะกล่าวถึงผลลัพธ์ที่ได้จากการออกแบบระบบบริหารจัดการซ่อมบำรุงรักษา
เครือข่ายในสนามบินของบริษัท สามารถคอมพิวเตอร์ จำกัด ในสนามบินสุวรรณภูมิ คณะผู้จัดทำได้
จัดเก็บผลการทดสอบระบบดังต่อไปนี้

4.1 การทดสอบแอปพลิเคชัน

4.1.1 ทดสอบใช้งานแอปพลิเคชันของหน้าต่าง Login

หน้าต่าง login ใช้สำหรับการเข้าสู่ระบบ แสดงดังรูปที่ 4.1 โดยก่อนจะเข้าสู่ระบบได้
นั้น จะต้องทำการบันทึกข้อมูล Username และ Password ลงในฐานข้อมูลก่อน โดยผู้ดูแลระบบจะ
มีค่าข้อมูล admin เท่ากับ 1 สำหรับผู้ใช้ระบบจะมีค่าข้อมูล admin เท่ากับ 0 แสดงดังรูปที่ 4.2

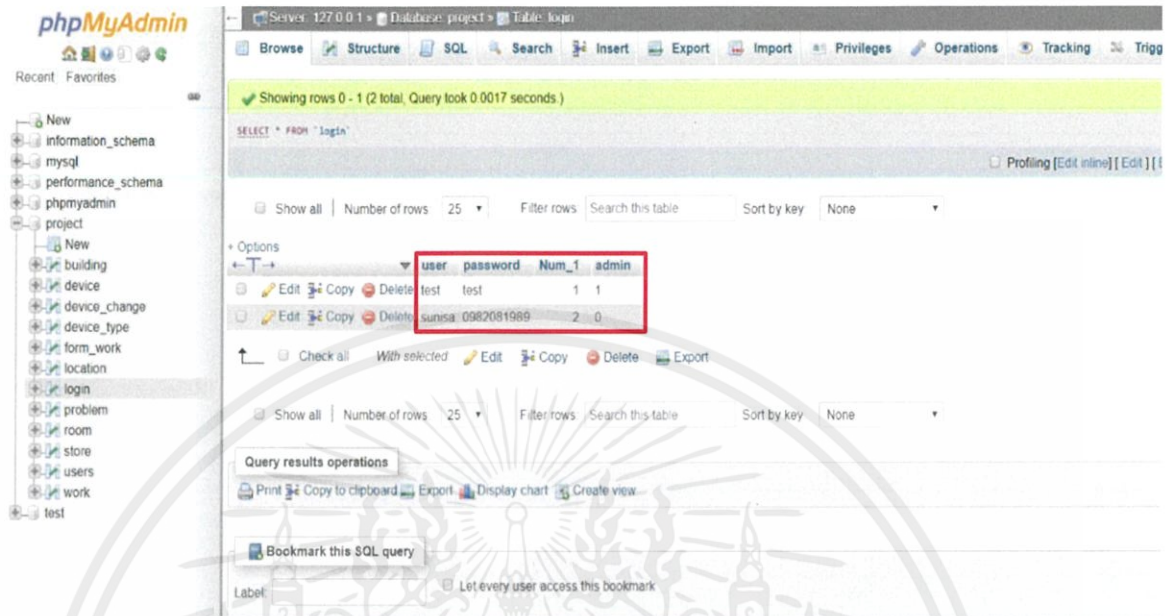


(ก)

(ข)

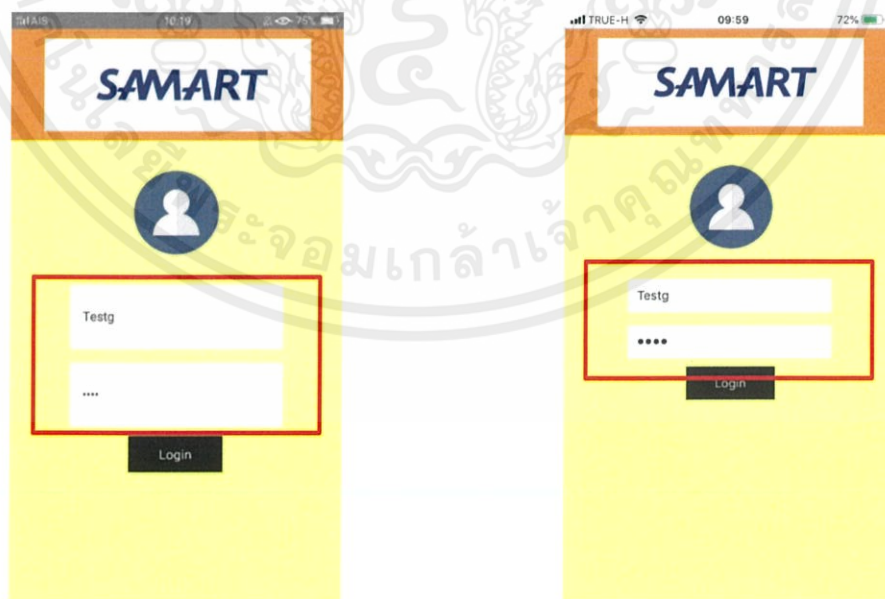
รูปที่ 4.1 หน้าต่าง Login (ก) Android และ (ข) iOS

เมื่อกรอกข้อมูล Username และ password ถูกต้องตามข้อมูลที่บันทึกไว้ในฐานข้อมูล ซึ่งแสดงตัวอย่างในรูปที่ 4.2 จะสามารถเข้าสู่หน้าต่างต่อไปได้



รูปที่ 4.2 ฐานข้อมูลตาราง “login”

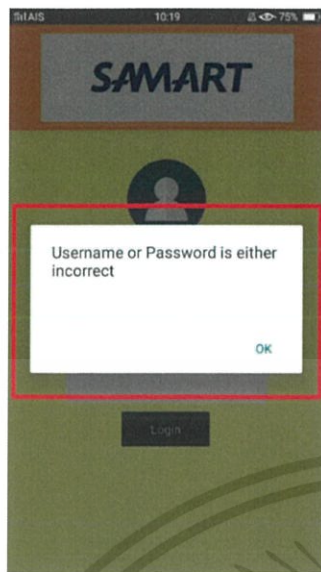
หากผู้ใช้งานกรอก Username หรือ Password ไม่ตรงตามข้อมูลที่บันทึกไว้ในฐานข้อมูล ระบบจะทำการแจ้งเตือนว่า “Username or Password either incorrec” ดังรูปที่ 4.3 และ 4.4 และจะไม่สามารถเข้าสู่หน้าต่างต่อไปได้



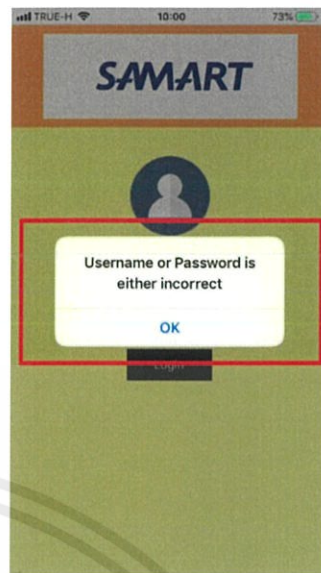
(ก)

(ข)

รูปที่ 4.3 หน้าต่าง Login เมื่อกรอก Username และ Password (ก) Android และ (ข) iOS



(ก)



(ข)

รูปที่ 4.4 การแจ้งเตือนในหน้าต่าง Login เมื่อกรอก Username และ Password

(ก) Android และ (ข) iOS

4.1.2 ทดสอบใช้งานแอปพลิเคชันของหน้าต่าง Preventive Maintenance

เมื่อเข้าสู่ระบบ (Login) สำเร็จจะพามาหน้าต่างหลักของแอปพลิเคชัน โดยมีระบบ Logout, PM (Preventive Maintenance), CM (Corrective Maintenance) และคลังพัสดุ แสดงดังรูปที่ 4.5 จากนั้นกดที่ปุ่ม “PM Preventive Maintenance” เพื่อเข้าใช้งานในส่วนของระบบนี้



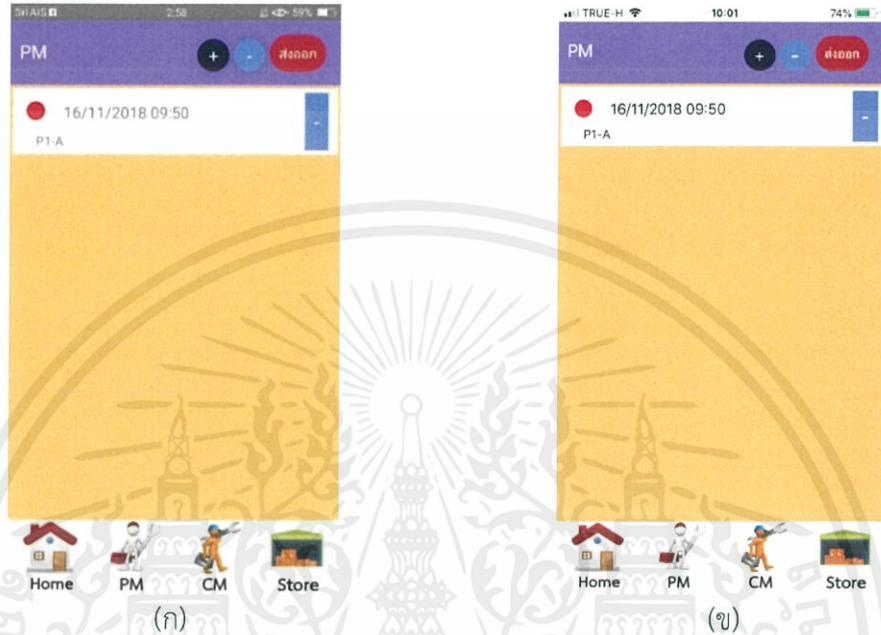
(ก)



(ข)

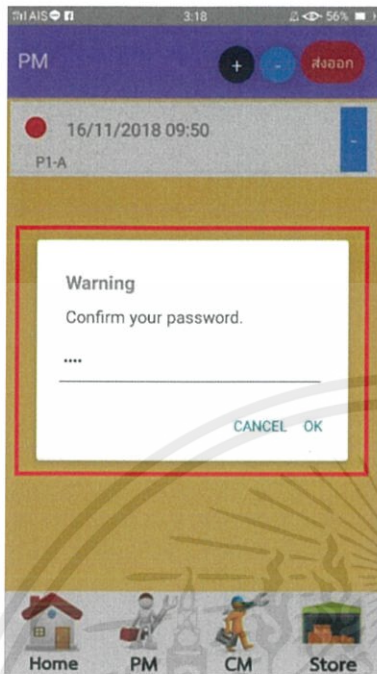
รูปที่ 4.5 หน้าต่าง Home ของแอปพลิเคชัน (ก) Android และ (ข) iOS

เมื่อทำการกดปุ่มตามรูปที่ 4.5 จะนำมาที่หน้าต่างดังรูปที่ 4.6 เป็นหน้าต่างสำหรับเพิ่มงานหรือลบงานและแสดงงานของแต่ละรอบในการทำ Preventive Maintenance โดยผู้ดูแลระบบจะมีสิทธิ์ในการเพิ่มงานโดยการกดปุ่ม “+” และมีสิทธิ์ในการลบงานโดยการกดปุ่ม “-” ด้านบนสุดของหน้าต่าง

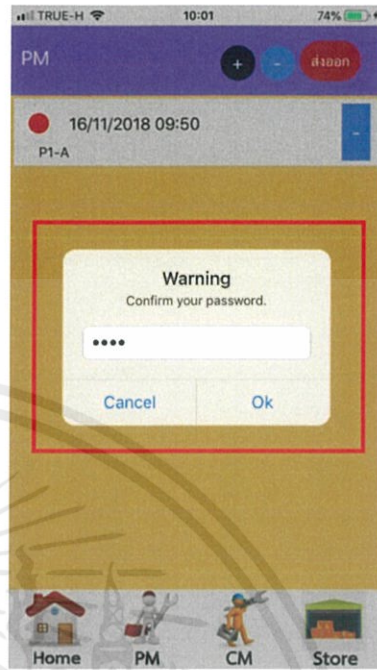


รูปที่ 4.6 หน้าต่างเพิ่ม ลบ และแสดงงานของ Preventive Maintenance
(ก) Android และ (ข) iOS

เมื่อกดเพิ่มงานที่ปุ่ม “+” จะต้องทำการใส่ Password ซึ่งเป็น Password ของผู้ดูแลระบบดังรูปที่ 4.7 เมื่อกรอก Password ถูกต้องจะนำไปหน้าเพิ่มงานดังรูปที่ 4.8 โดยมีให้เลือกวัน เวลา และสถานที่ดังรูปที่ 4.9 – 4.11 หลังจากเลือกวัน เวลา และสถานที่ ให้กดปุ่ม submit จะขึ้นแจ้งเตือนว่า “Insert successfully” ดังรูปที่ 4.12

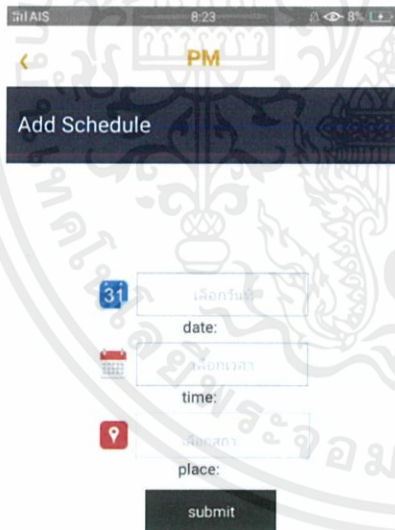


(ก)

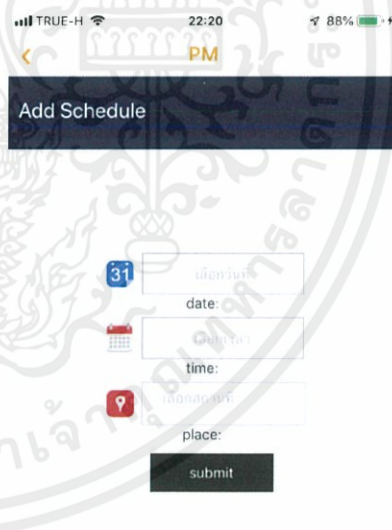


(ข)

รูปที่ 4.7 หน้าต่างกรอก Password เพื่อเพิ่มงานสำหรับผู้ดูแลระบบ (ก) Android และ (ข) iOS

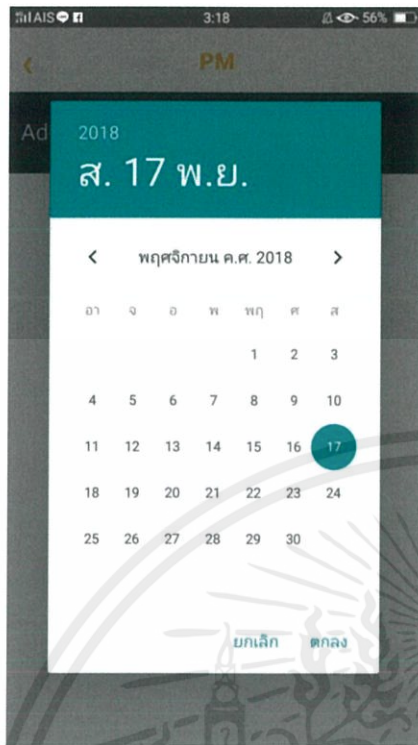


(ก)

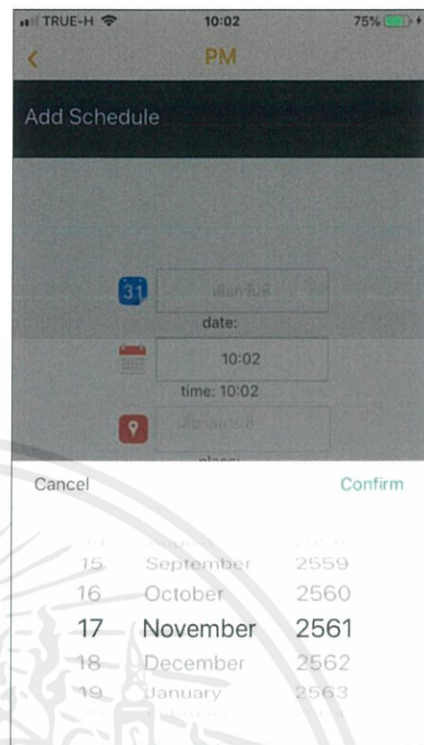


(ข)

รูปที่ 4.8 หน้าต่างเพิ่มงาน (Add Schedule) (ก) Android และ (ข) iOS

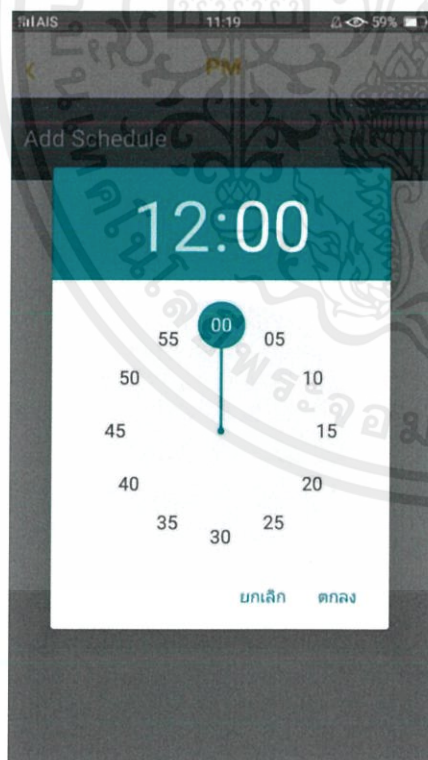


(ก)

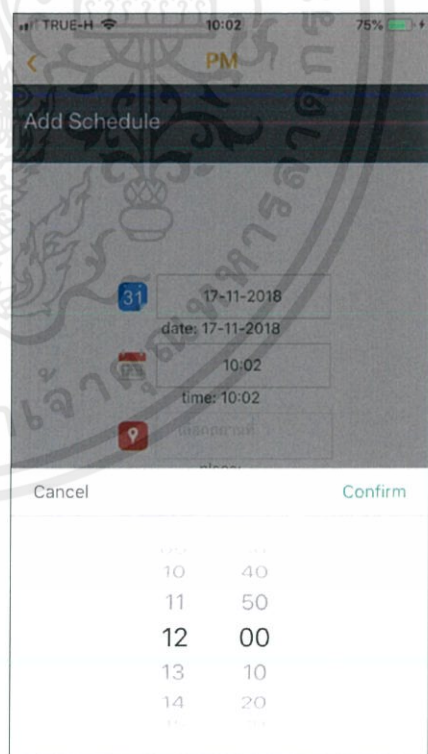


(ข)

รูปที่ 4.9 หน้าต่างเพิ่มวันที่ (ก) Android และ (ข) iOS



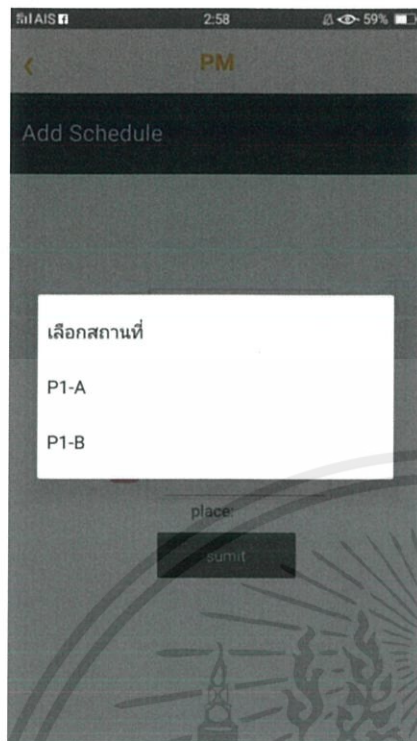
(ก)



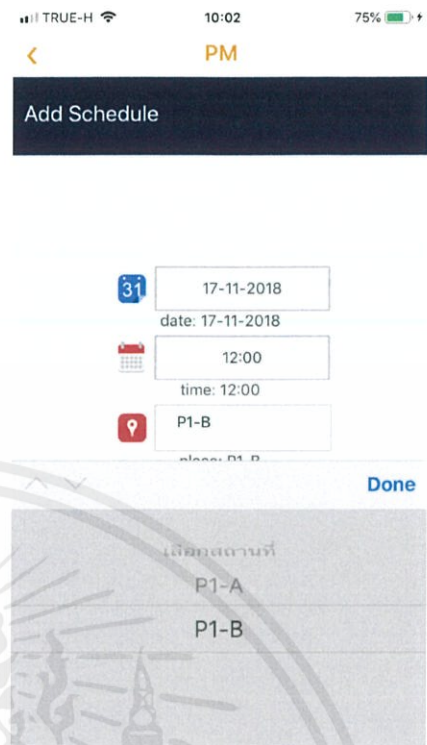
(ข)

รูปที่ 4.10 หน้าต่างเพิ่มเวลา (ก) Android และ (ข) iOS

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

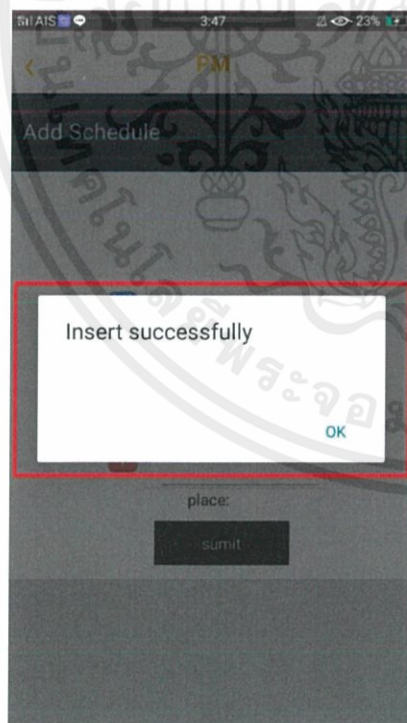


(ก)

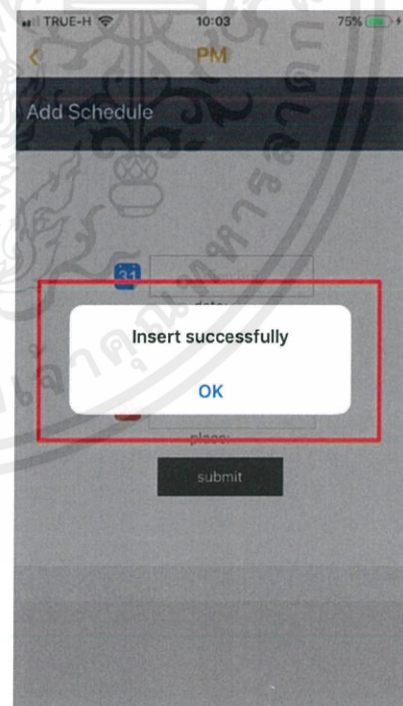


(ข)

รูปที่ 4.11 หน้าต่างเพิ่มสถานที่ (ก) Android และ (ข) iOS



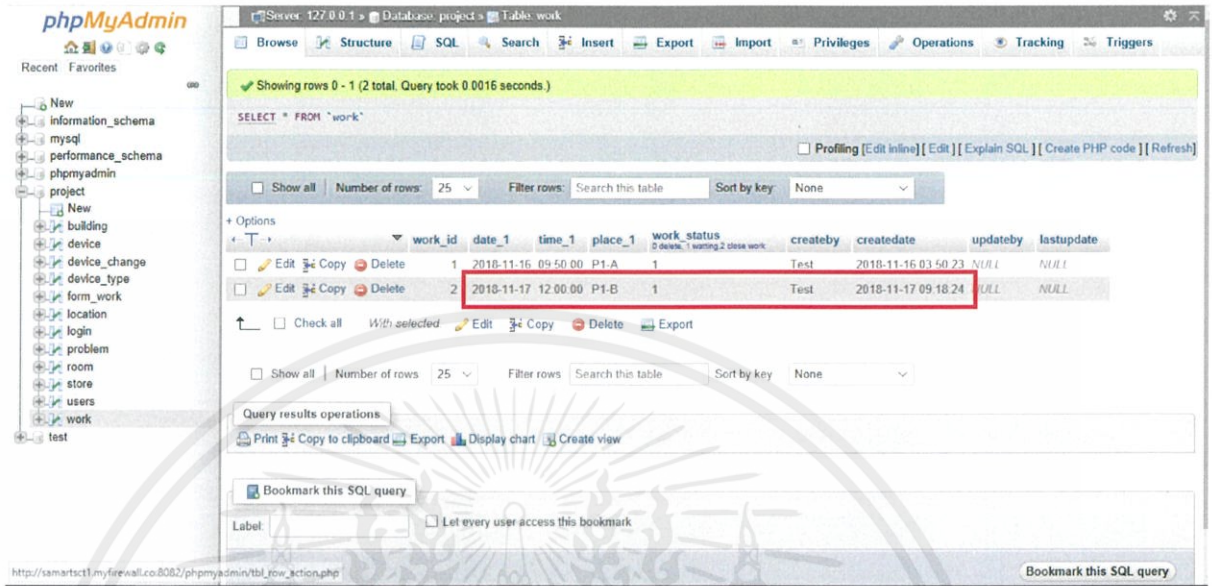
(ก)



(ข)

รูปที่ 4.12 หน้าต่างแจ้งเตือนเมื่อเพิ่มงานสำเร็จ (ก) Android และ (ข) iOS

เมื่อเพิ่มข้อมูลสำเร็จแล้วข้อมูลที่กรอกจะเข้าสู่ฐานข้อมูลตามรูปที่ 4.13 และแสดงผลการเพิ่มงานในหน้าต่างดังรูปที่ 4.14

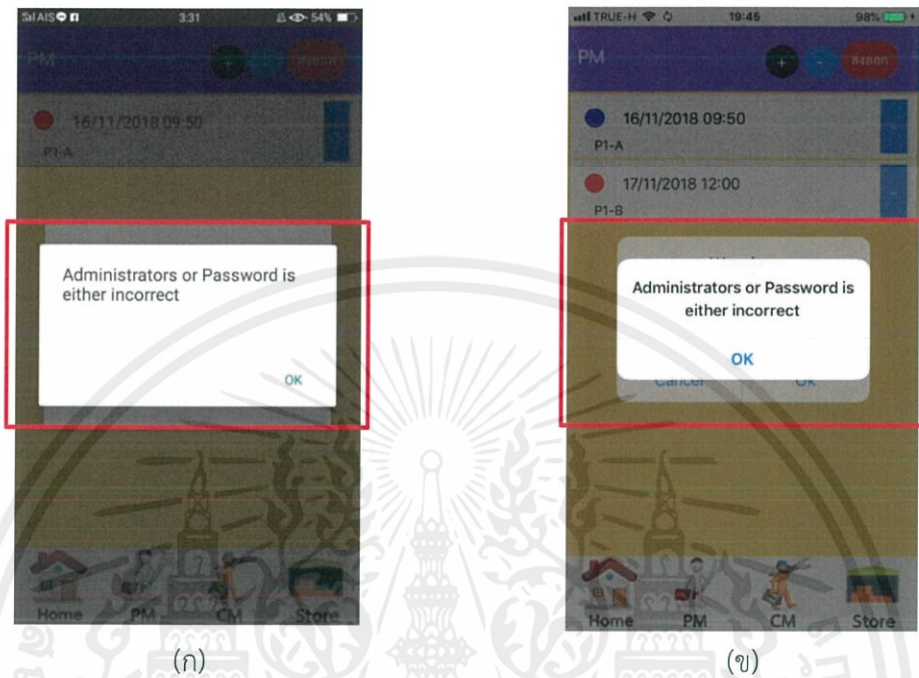


รูปที่ 4.13 ฐานข้อมูลตาราง “work” ที่ถูกบันทึกข้อมูลการเพิ่มงาน



รูปที่ 4.14 หน้าต่างแสดงงานที่ต้องทำเมื่อเพิ่มงานสำเร็จ (ก) Android และ (ข) iOS

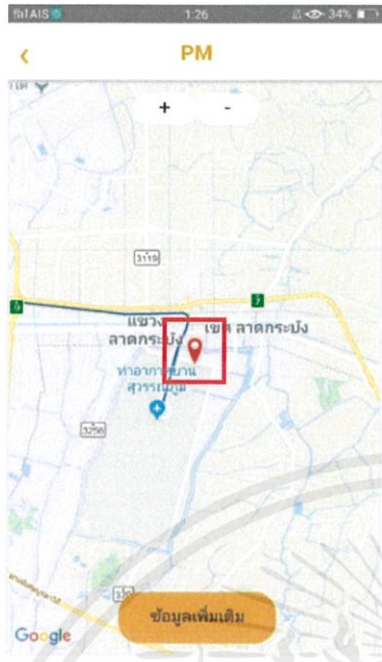
หากกรอก Password ไม่ตรงตามข้อมูลที่บันทึกไว้ในฐานข้อมูล ระบบจะทำการแจ้งเตือนว่า “Administrators or Password either incorrect” ดังรูปที่ 4.15 และจะไม่สามารถเข้าสู่หน้าต่างเพิ่มงานได้



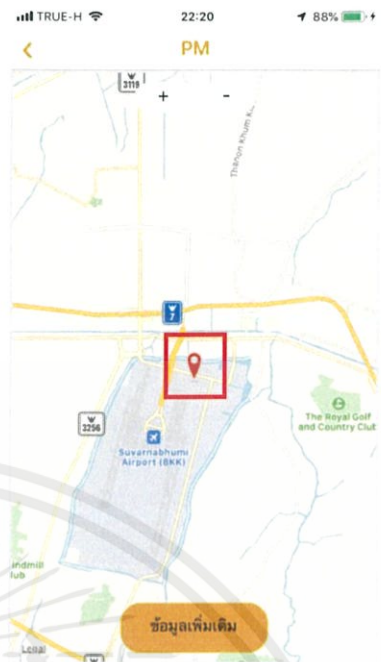
รูปที่ 4.15 หน้าต่างแจ้งเตือนเมื่อผู้ดูแลระบบกรอก Password ไม่ถูกต้อง

(ก) Android และ (ข) iOS

เมื่อทำการเพิ่มงานจนครบที่ผู้ดูแลระบบต้องการ ต่อจากนั้นเมื่อผู้ใช้งานพร้อมปฏิบัติงานไปที่หน้างานเพื่อทำ Preventive Maintenance จะทำการรดงานที่ต้องการไปทำในหน้าแสดงงาน PM ตามรูปที่ 4.14 นำไปสู่หน้าต่างที่ถูกปิดกั้นด้วยคำพิกัดสถานที่ที่ต้องเข้าปฏิบัติงานบนแผนที่แสดงในรูปที่ 4.16 เมื่อกดบนหมุดสีแดงจะมีปุ่มให้กดไปเปิดที่ Google Maps และนำทางโดยใช้ Google Maps Application แสดงในรูปที่ 4.16 – 4.20

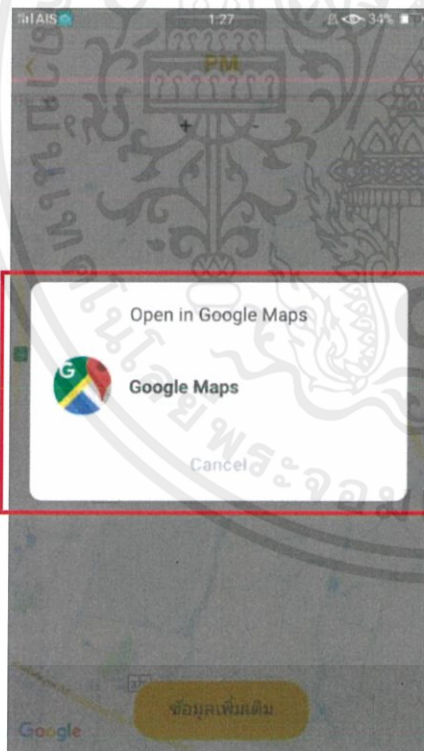


(ก)

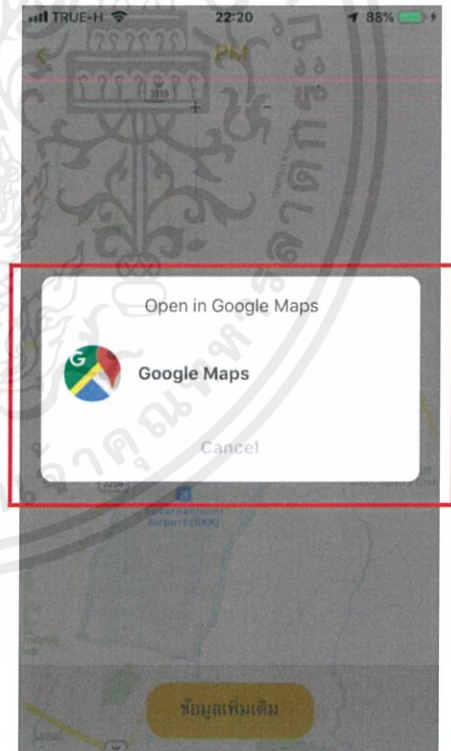


(ข)

รูปที่ 4.16 หน้าต่างปักหมุดสถานที่บนแผนที่ (ก) Android และ (ข) iOS



(ก)

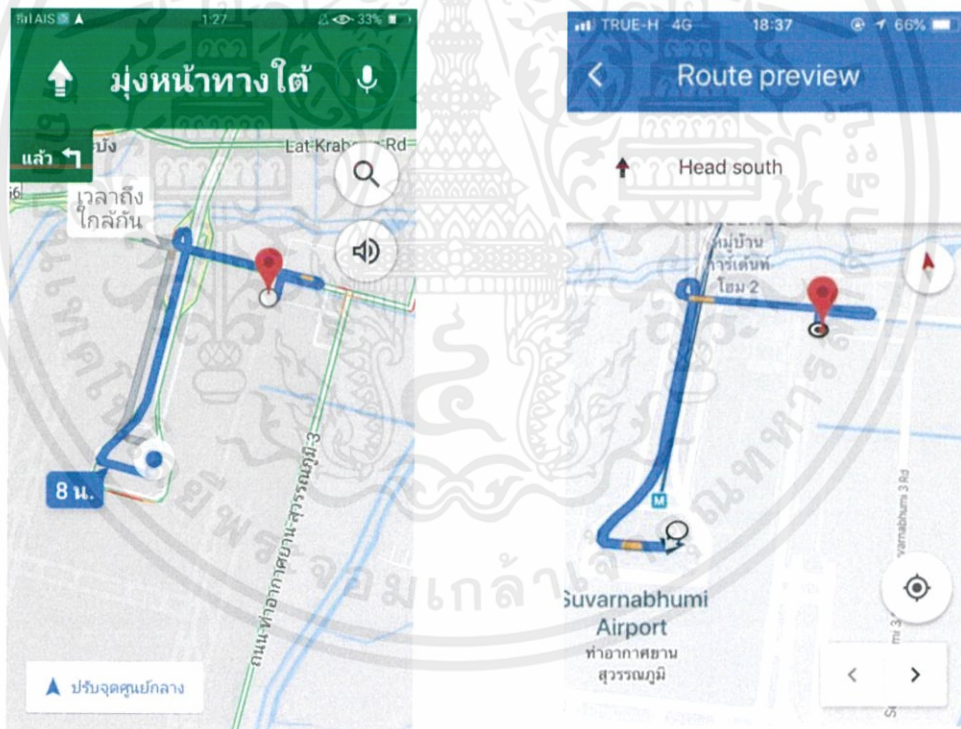


(ข)

รูปที่ 4.17 หน้าต่างให้กดไปเปิดที่ Google Maps (ก) Android และ (ข) iOS



รูปที่ 4.18 ทดสอบใช้งานแอปพลิเคชันเพื่อกดไปเปิดที่ Google Maps



(ก)

(ข)

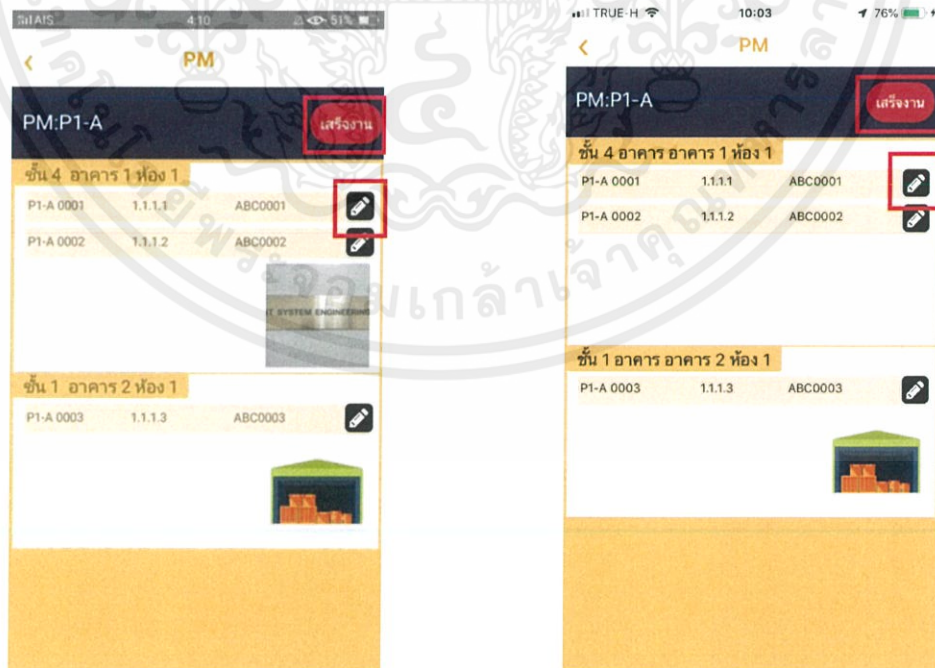
รูปที่ 4.19 หน้าต่างการใช้ Google Maps Application เพื่อนำทาง

(ก) Android และ (ข) iOS



รูปที่ 4.20 ทดสอบการใช้ Google Maps Application เพื่อนำทาง

เมื่อถึงสถานที่ที่ต้องการให้กลับมาที่หน้าต่างที่ปกคลุมสถานที่ดังรูปที่ 4.16 เพื่อมากรอกข้อมูลแบบฟอร์มการบำรุงรักษาโดยกดปุ่มข้อมูลเพิ่มเติม จะแสดงหน้าต่างเกี่ยวกับข้อมูลบอกถึงสถานที่ ชั้น อาคาร ห้อง อุปกรณ์ที่ติดตั้งที่สถานะนั้น ๆ รวมถึงเลขไอพีแอดเดรส และ Serial Number ของอุปกรณ์พร้อมทั้งรูปหน้าห้องของห้องนั้น ๆ และทำการกดปุ่มด้านข้างชื่ออุปกรณ์ เพื่อเข้าไปที่ตัวอุปกรณ์ดังรูปที่ 4.21 เพื่อไปที่หน้าต่างการกรอกข้อมูลบำรุงรักษาเครือข่ายแสดงในรูปที่ 4.22

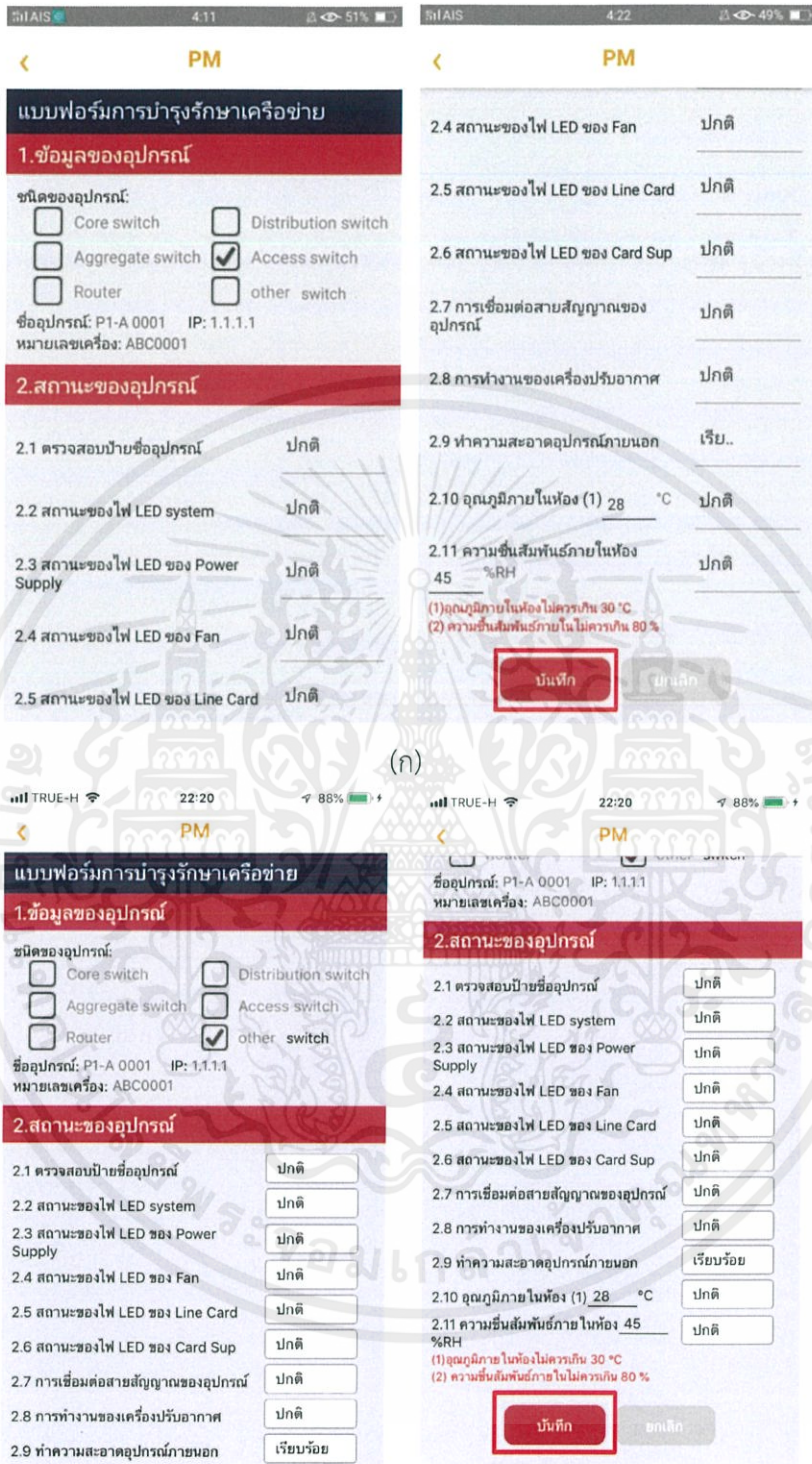


(ก)

(ข)

รูปที่ 4.21 หน้าต่างแสดงข้อมูลเพิ่มของสถานที่นั้น ๆ (ก) Android และ (ข) iOS

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



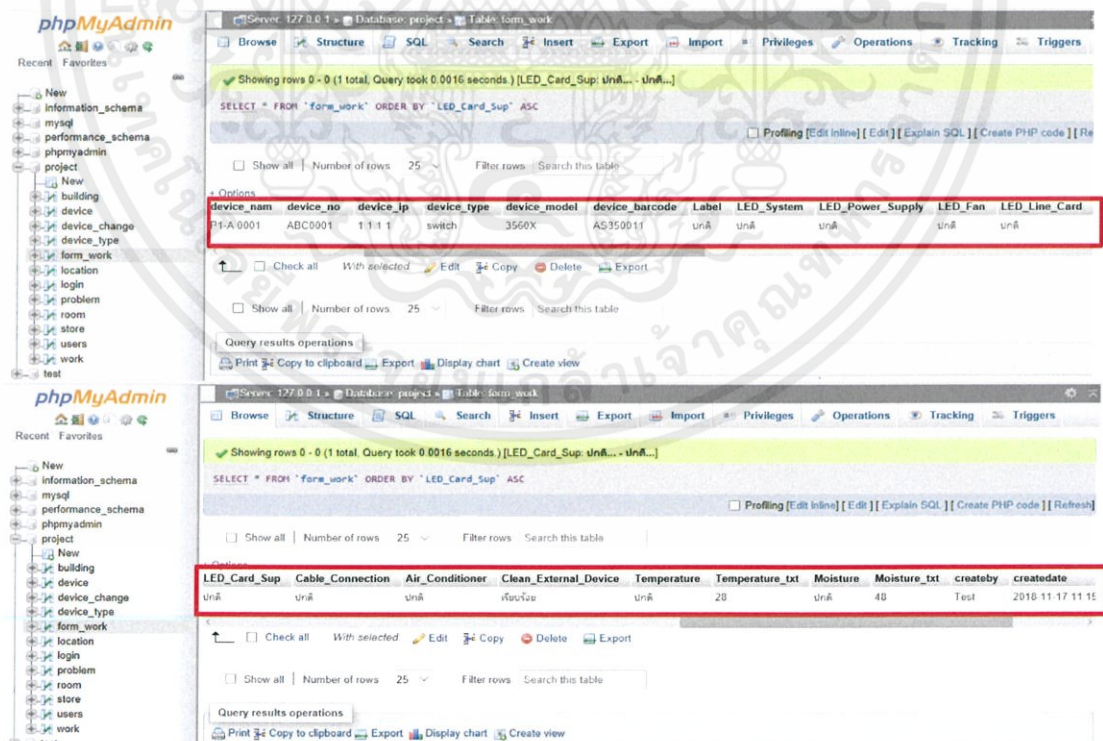
(ข)

รูปที่ 4.22 หน้าต่างแบบฟอร์มการบำรุงรักษาเครือข่าย (ก) Android และ (ข) iOS



รูปที่ 4.23 ทดสอบการใช้แอปพลิเคชันเพื่อกรอกแบบฟอร์มการบำรุงรักษาเครือข่ายและบันทึกข้อมูล

หลังจากตรวจสอบอุปกรณ์ ทำความสะอาดอุปกรณ์ จากนั้นทำการกรอกข้อมูลในหน้าต่างแบบฟอร์มการบำรุงรักษาเครือข่ายแสดงดังรูปที่ 4.23 และทำการบันทึกข้อมูลตามรูปที่ 4.22 และข้อมูลจะถูกจัดเก็บลงในฐานข้อมูลตามรูปที่ 4.24

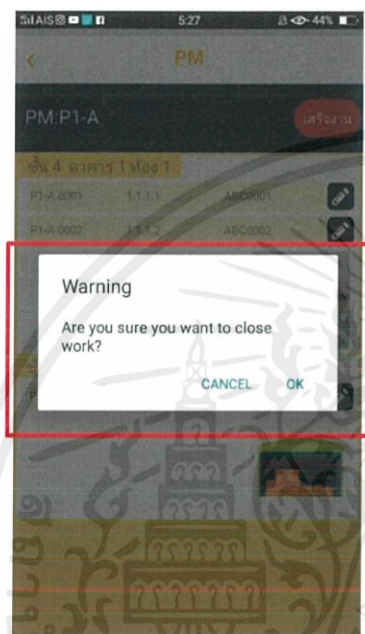


รูปที่ 4.24 ฐานข้อมูลตาราง “from_work” ที่ถูกบันทึกข้อมูล

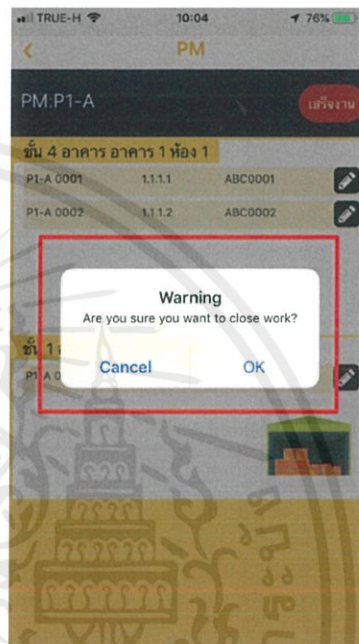
การกรอกแบบฟอร์มการบำรุงรักษาเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

หลังจากกรอกข้อมูลครบถ้วนทุกอุปกรณ์ให้ทำการกดปุ่ม “เสร็จงาน” ตามรูปที่ 4.21 เพื่อเปลี่ยนสถานะของงานนั้นโดยเมื่อกดปุ่มจะมีการแจ้งเตือนว่า “Are you sure want to close work?” ดังรูปที่ 4.25 เมื่อกด OK จะแจ้งว่า “Close work successfully” ดังรูปที่ 4.26 และเมื่อกลับไปหน้าแสดงงานจะเปลี่ยนสถานะจากสีแดงเป็นสีฟ้าดังรูปที่ 4.27 ซึ่ง “สีแดง” แสดงสถานะงานยังไม่เสร็จและ “สีฟ้า” แสดงสถานะของงานที่เสร็จแล้ว

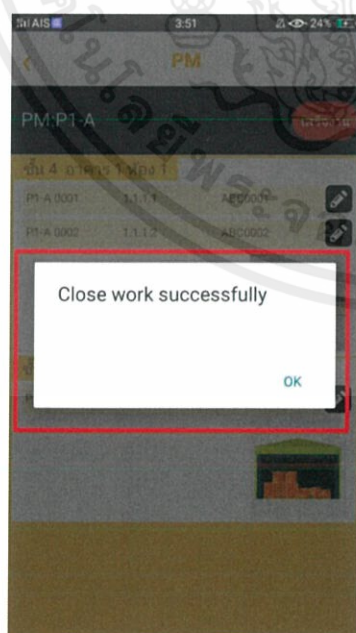


(ก)

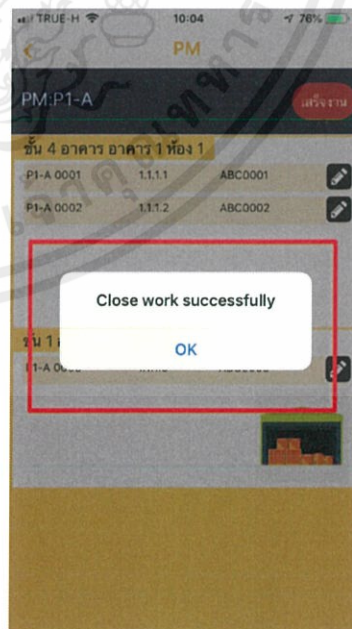


(ข)

รูปที่ 4.25 หน้าต่างแจ้งเตือนถามเมื่อกดปุ่ม “เสร็จงาน” (ก) Android และ (ข) iOS

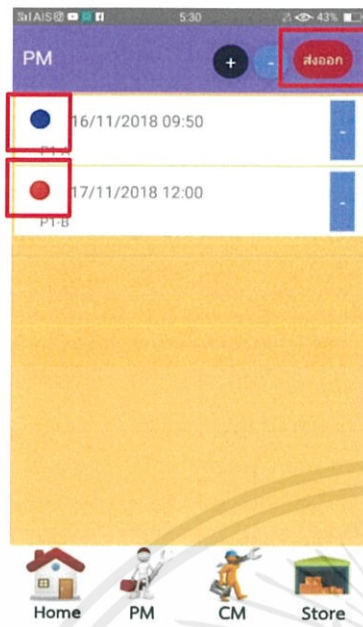


(ก)

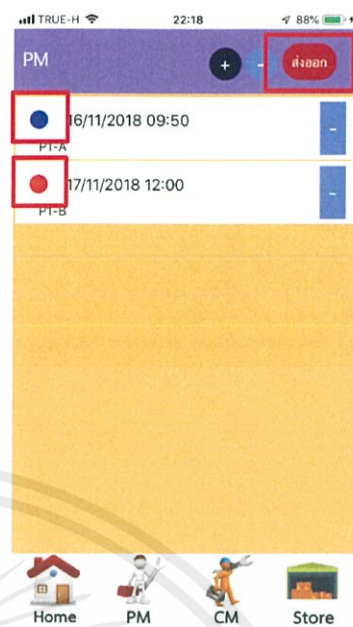


(ข)

รูปที่ 4.26 หน้าต่างแจ้งเตือนเมื่อปิดงานสำเร็จแล้ว (ก) Android และ (ข) iOS



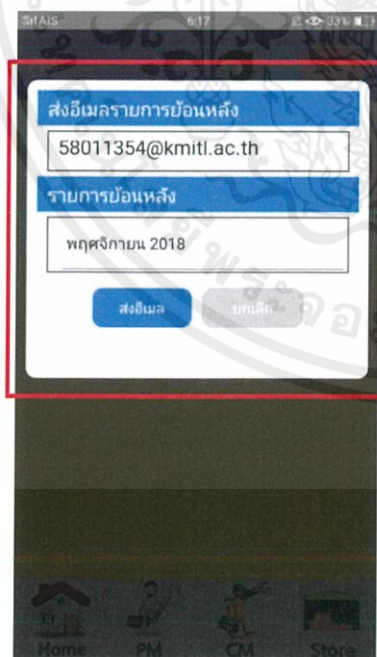
(ก)



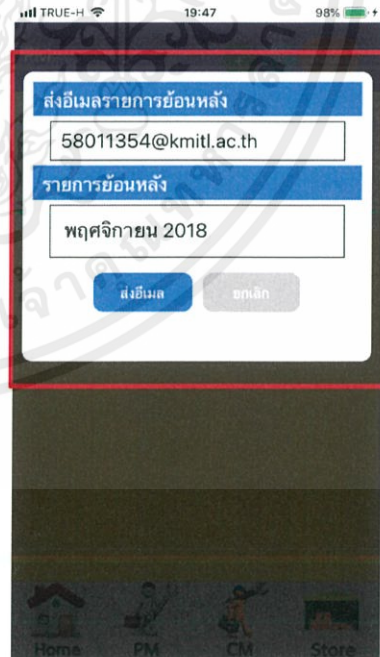
(ข)

รูปที่ 4.27 หน้าต่างแสดงงานที่ต้องทำเมื่อปิดงานสำเร็จ (ก) Android และ (ข) iOS

เมื่อต้องการนำข้อมูลจากฟอร์มการบำรุงรักษาเครือข่ายและสถานะของงานออกมาเป็นไฟล์ Excel สามารถทำได้โดยการกดปุ่มส่งออกตามรูปที่ 4.27 จะขึ้นหน้าต่างที่สามารถส่งออกข้อมูลไปทางอีเมลตามรูปที่ 4.28 ทำการกรอกอีเมลที่ต้องการส่งและเลือกเดือนที่ต้องการ เมื่อส่งข้อมูลสำเร็จจะแจ้งเตือนว่า “Send email successfully” แสดงดังรูปที่ 4.29

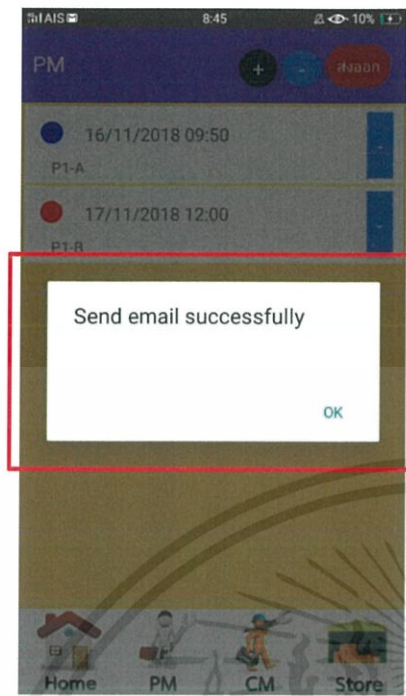


(ก)

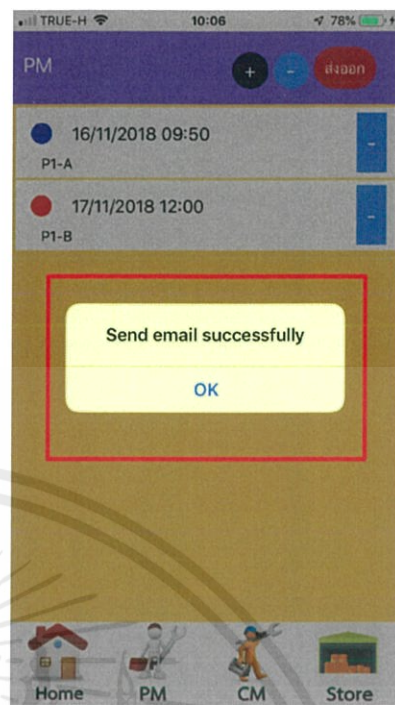


(ข)

รูปที่ 4.28 หน้าต่างกรอกอีเมลและเลือกเดือน (ก) Android และ (ข) iOS



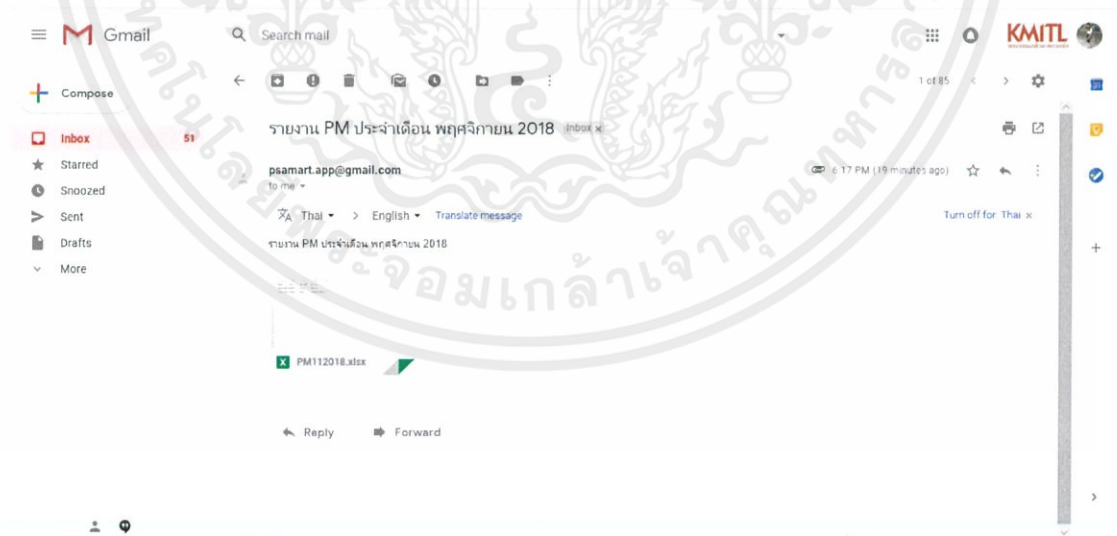
(ก)



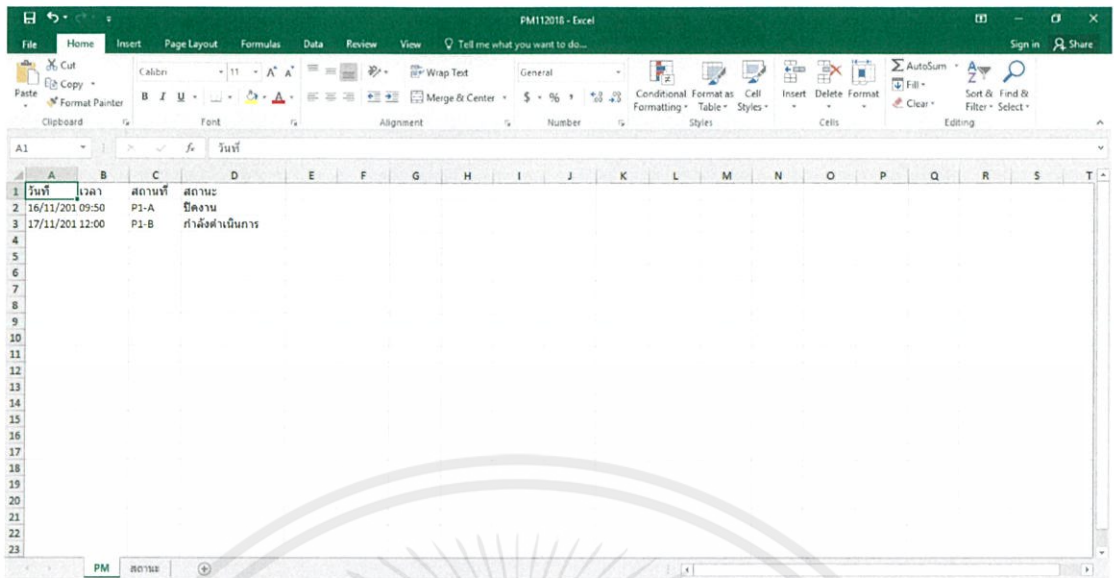
(ข)

รูปที่ 4.29 หน้าต่างแจ้งเตือนเมื่อส่งอีเมลสำเร็จ (ก) Android และ (ข) iOS

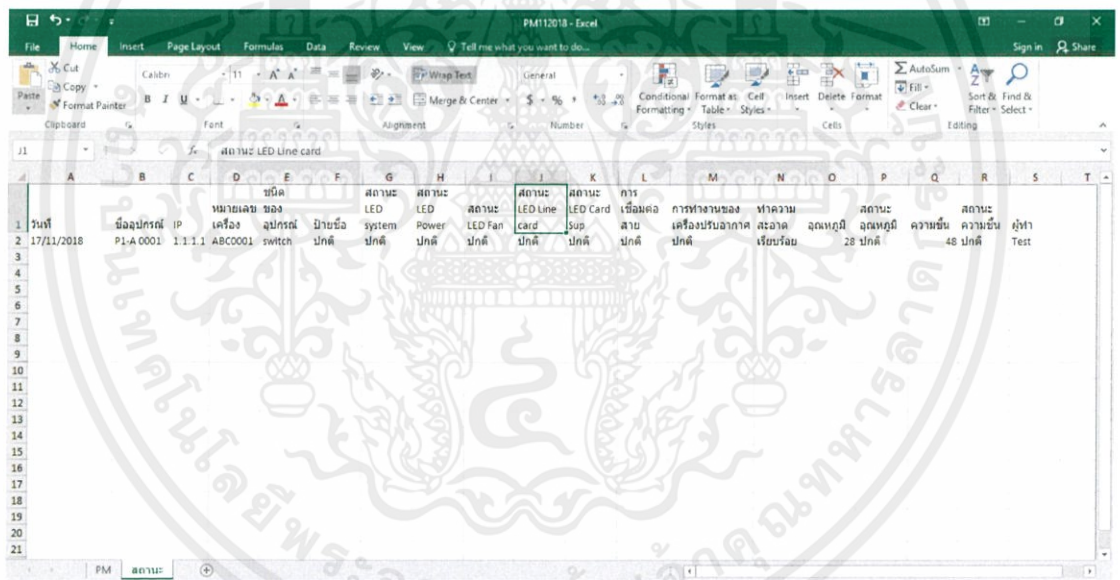
จากนั้นทำการเปิดข้อความในอีเมลจะพบไฟล์ที่ส่งมาดังรูปที่ 4.30 เมื่อเปิดไฟล์ Excel ในอีเมลดังรูปที่รูปที่ 4.31 จะเป็นข้อมูลการแสดงสถานะของงาน และรูปที่ 4.32 เป็นการแสดงข้อมูลแบบฟอร์มการบำรุงรักษาเครือข่าย



รูปที่ 4.30 หน้าต่างแสดงข้อความรายงาน PM ในอีเมล



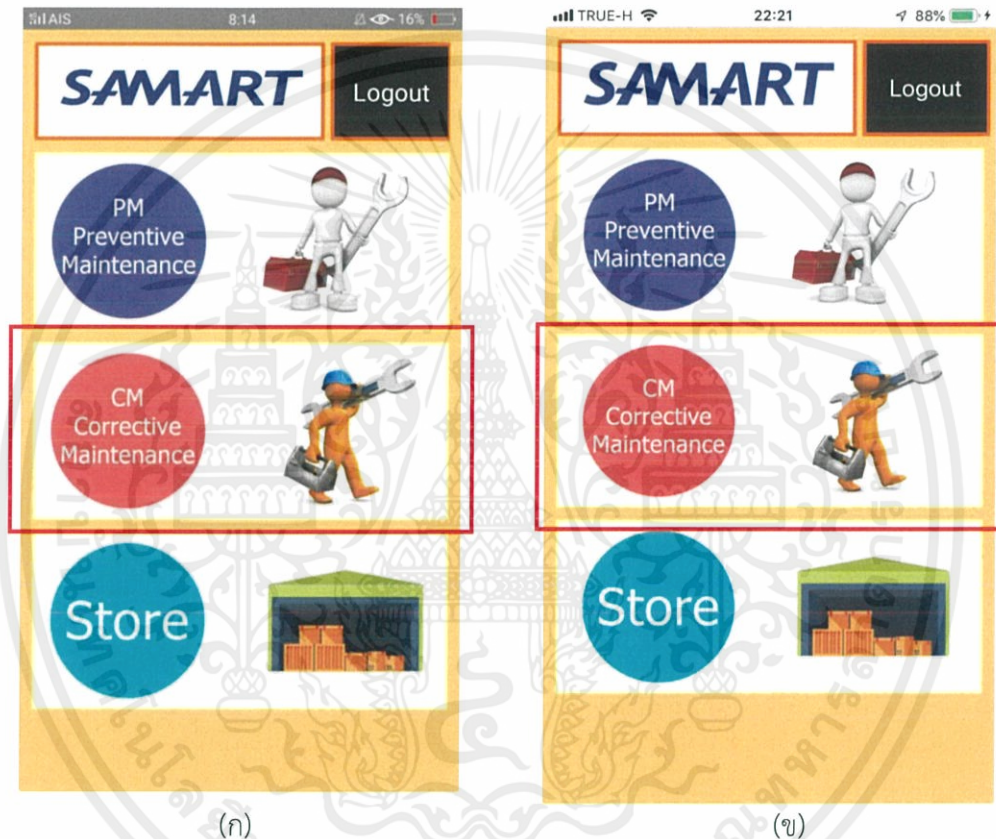
รูปที่ 4.31 หน้าต่างโปรแกรม Excel แสดงสถานะของงาน PM



รูปที่ 4.32 หน้าต่างโปรแกรม Excel แสดงรายงานผลจากแบบฟอร์มการบำรุงรักษาเครือข่าย

4.1.3 ทดสอบใช้งานแอปพลิเคชันของหน้าต่าง Corrective Maintenance

เมื่ออุปกรณ์มีปัญหาจะทำการปฏิบัติงาน Corrective Maintenance (CM) โดยเมื่อเปิดแอปพลิเคชัน และ Login สำเร็จ ให้เลือกกดปุ่ม CM ดังรูป 4.33 จากนั้นจะนำไปหน้าหลักของปฏิบัติงาน CM แสดงดังรูปที่ 4.34 โดยจะมีวิธีการเตรียมเข้าปฏิบัติงานเบื้องต้น การนำเส้นทางไปจุดที่อุปกรณ์มีปัญหา แบบฟอร์มสำหรับการแจ้งเปลี่ยนอุปกรณ์ รวมทั้งสามารถดูย้อนหลังปัญหาที่เกิดขึ้น



รูปที่ 4.33 หน้าต่าง Home ของแอปพลิเคชัน (ก) Android และ (ข) iOS



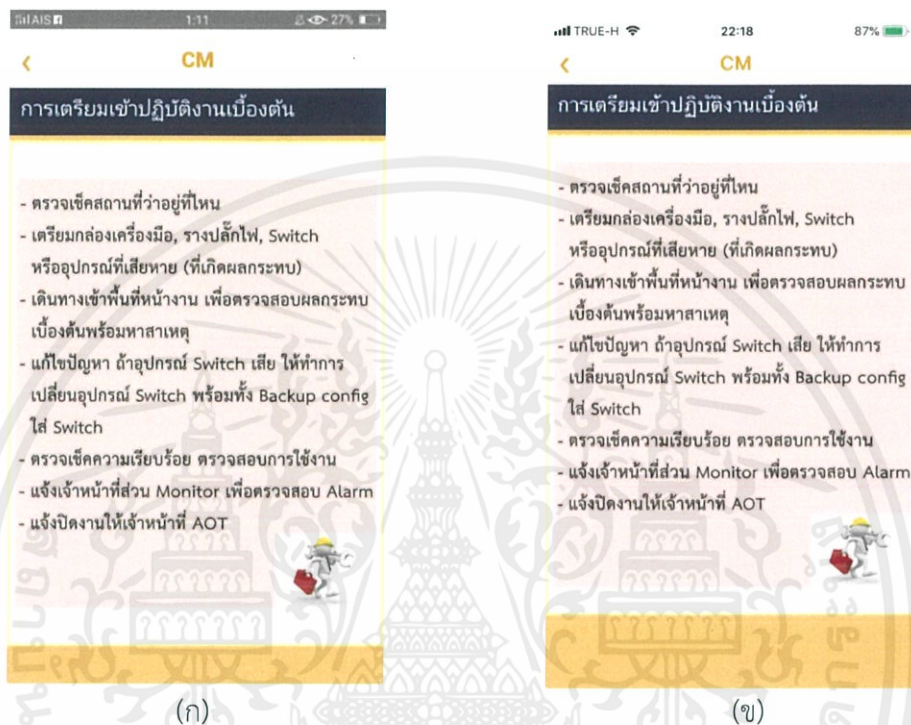
(ก)



(ข)

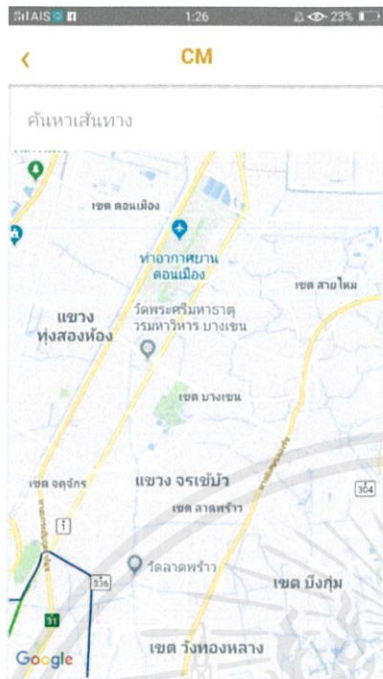
รูปที่ 4.34 หน้าต่างหลักของ Corrective Maintenance (ก) Android และ (ข) iOS

เริ่มต้นทดสอบเข้าไปดูในรายละเอียดขั้นตอนการปฏิบัติงาน โดยกดปุ่มการเตรียมเข้าปฏิบัติงานเบื้องต้นจะนำมาหน้ารายละเอียดขั้นตอนเบื้องต้น ดังรูปที่ 4.35 ให้ตรวจสอบตามขั้นตอนและเตรียมของเข้าพื้นที่ก่อนเข้าพื้นที่ปฏิบัติงานจริง

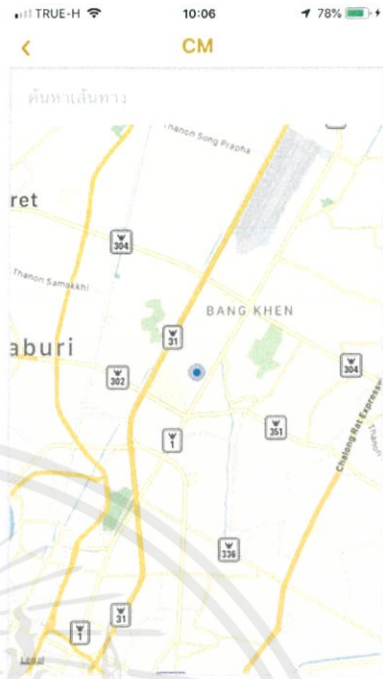


รูปที่ 4.35 หน้าต่างการเตรียมเพื่อเข้าปฏิบัติงานเบื้องต้น (ก) Android และ (ข) iOS

เมื่อรู้พื้นที่ที่อุปกรณ์มีปัญหาสามารถค้นหาเส้นทางได้โดยกดปุ่ม “เส้นทาง” ดังรูปที่ 4.34 จะนำไปหน้าต่างค้นหาเส้นทางดังรูปที่ 4.36 สามารถค้นหาพื้นที่ที่ต้องการไป เมื่อกดเลือกพื้นที่ที่ต้องการจะนำไปหน้าปักหมุดบนแผนที่ดังรูปที่ 4.37 และสามารถนำไปยังจุดนั้นได้ จากการกดหมุดสีแดงบนแผนที่จะค่าพิกัดไปใช้ใน Google Maps และใช้ Google Maps ในการนำทาง



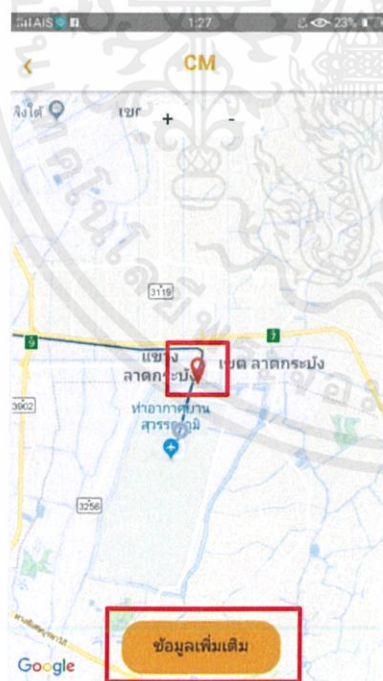
(ก)



(ข)

รูปที่ 4.36 หน้าต่างค้นหาเส้นทางเพื่อเข้าปฏิบัติงาน Corrective Maintenance

(ก) Android และ (ข) iOS



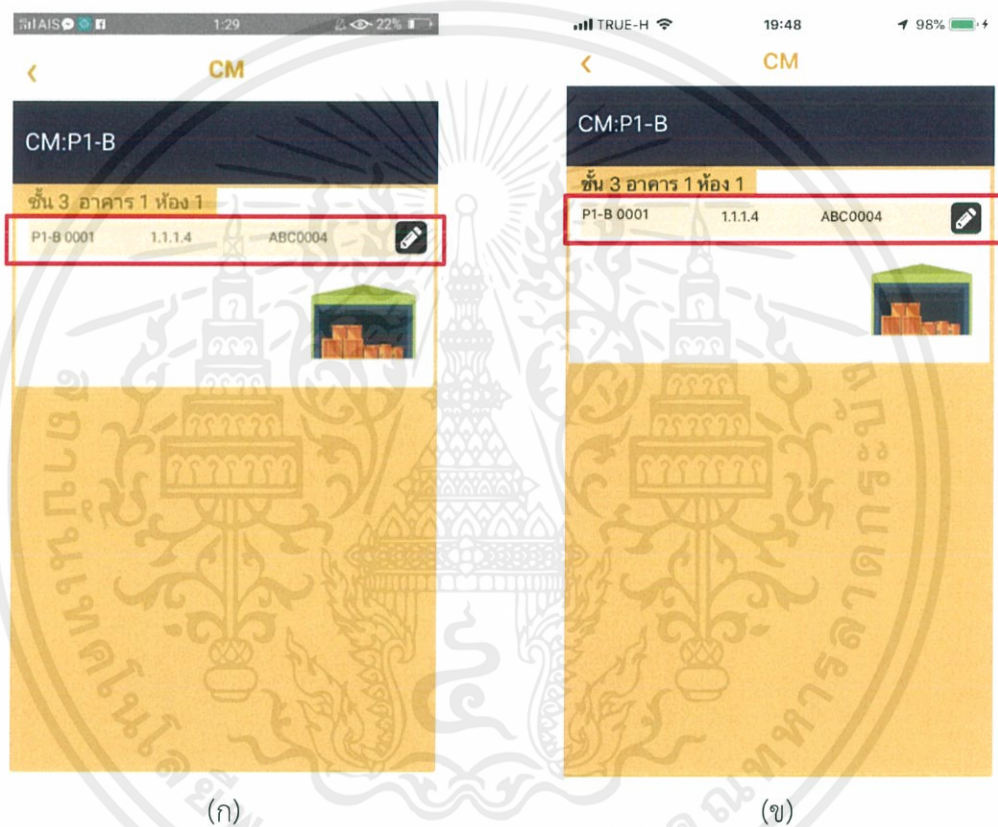
(ก)



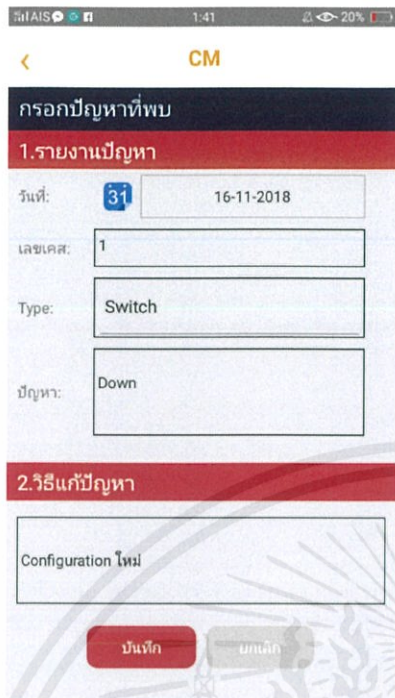
(ข)

รูปที่ 4.37 หน้าต่างปิกหมุดสถานที่บนแผนที่ (ก) Android และ (ข) iOS

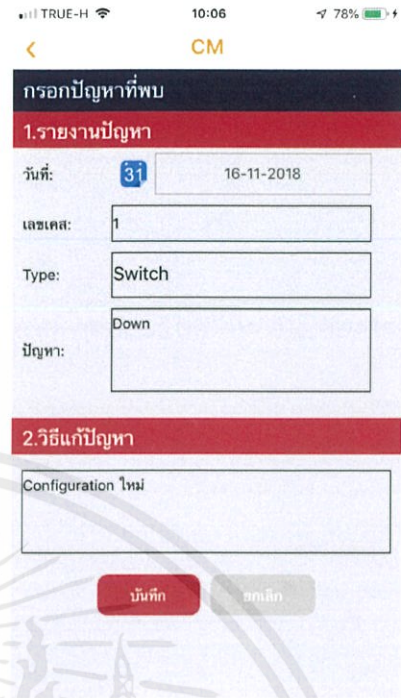
หลังจากไปถึงสถานที่และแก้ไขอุปกรณ์ที่มีปัญหาเสร็จแล้ว ให้กดปุ่ม “ข้อมูลเพิ่มเติม” แสดงในรูปที่ 4.37 จะนำไปหน้ารายละเอียดอุปกรณ์ของสถานที่นั้น ซึ่งประกอบด้วยชื่ออุปกรณ์ เลข ไอพีแอดเดรส และ Serial Number รวมถึงรูปหน้าห้องที่มีอุปกรณ์อยู่ของอุปกรณ์ในสถานที่นั้นและ กดปุ่มที่อุปกรณ์มีปัญหาแสดงในรูปที่ 4.35 เพื่อกรอกรายละเอียดปัญหาที่เกิดขึ้นมีวันที่เกิดปัญหา เลขเคส type ปัญหา และวิธีแก้ปัญหาลงในรูปที่ 4.39 ทำการกรอกข้อมูลให้ครบถ้วนและกดปุ่ม บันทึก เมื่อบันทึกสำเร็จจะแจ้งเตือนว่า “input successfully” ตามรูปที่ 4.40 ข้อมูลจะถูกบันทึกจะถูกเก็บไว้ที่ฐานข้อมูลตามรูปที่ 4.41



รูปที่ 4.38 หน้าต่างข้อมูลเพิ่มเติมของสถานที่นั้น ๆ (ก) Android และ (ข) iOS

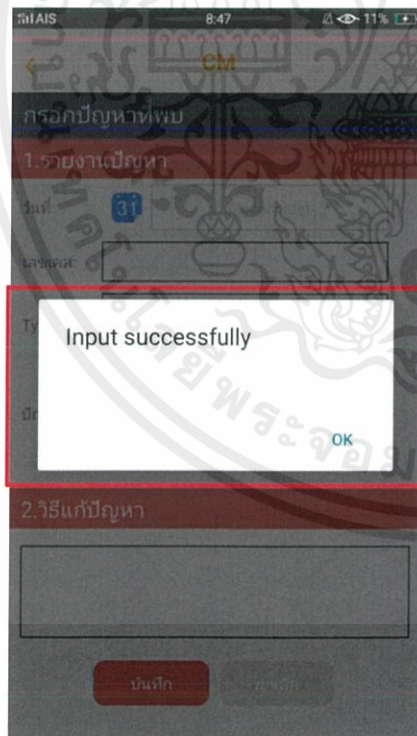


(ก)

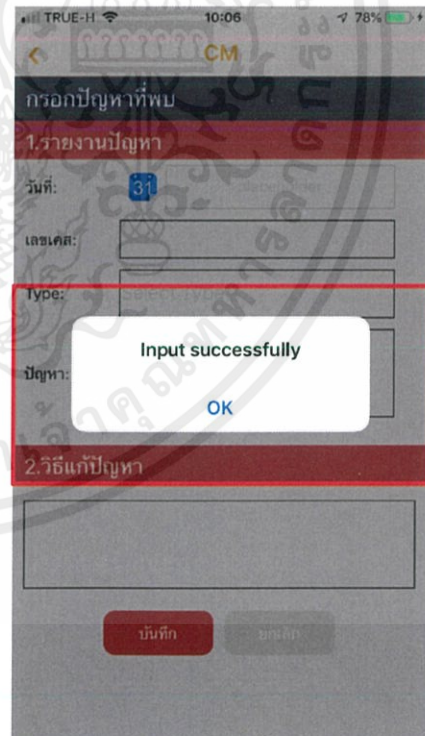


(ข)

รูปที่ 4.39 หน้าต่างกรอกปัญหาที่พบ (ก) Android และ (ข) iOS



(ก)



(ข)

รูปที่ 4.40 หน้าต่างแจ้งเตือนเมื่อกรอกปัญหาที่พบสำเร็จ (ก) Android และ (ข) iOS

device_name	device_no	device_ip	device_type	device_model	device_barcode	problem_date	problem_no	problem_type	problem_detail	problem_edit	createby	createdate
P1 B 0001	ABC0004	1.1.1.4	switch	3500X	AS350014	2018-11-16	1	Switch	Down	Configuration Issue	Test	2018-11-16 07:41

รูปที่ 4.41 ฐานข้อมูลตาราง “problem” ที่ถูกบันทึกข้อมูลปัญหาที่พบ

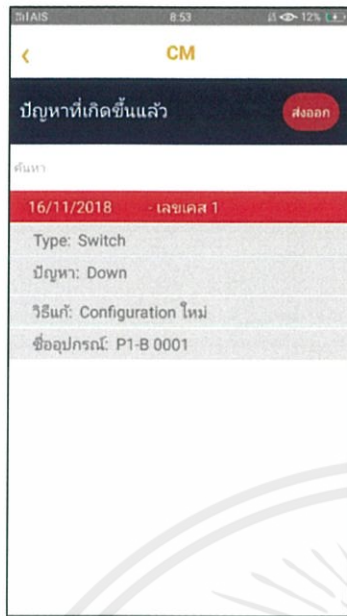
เมื่อต้องการดูรายละเอียดย้อนหลังปัญหาที่เกิดขึ้น สามารถทำได้โดยกดปุ่มปัญหาที่เกิดขึ้นดังรูปที่ 4.42 ที่หน้าต่างหลักของ CM จะนำไปหน้าต่างปัญหาที่เกิดขึ้น และสามารถกดค้นหาได้ตามดังรูปที่ 4.43



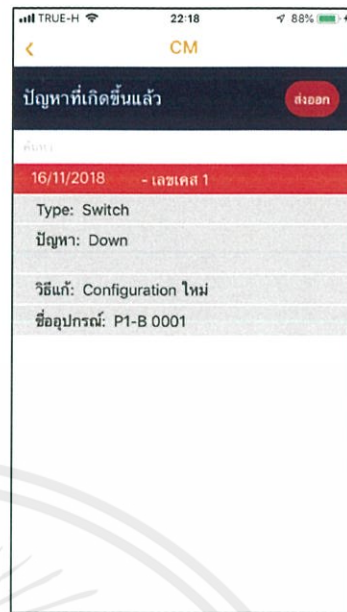
(ก)

(ข)

รูปที่ 4.42 หน้าต่างหลักของ Corrective Maintenance (ก) Android และ (ข) iOS



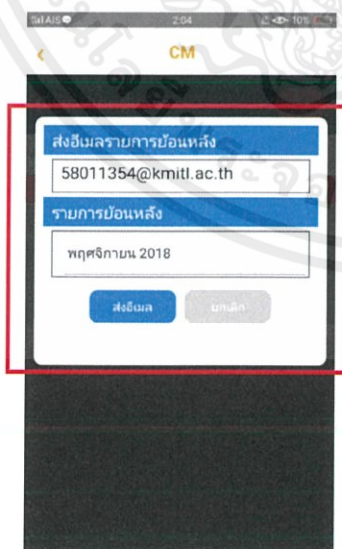
(ก)



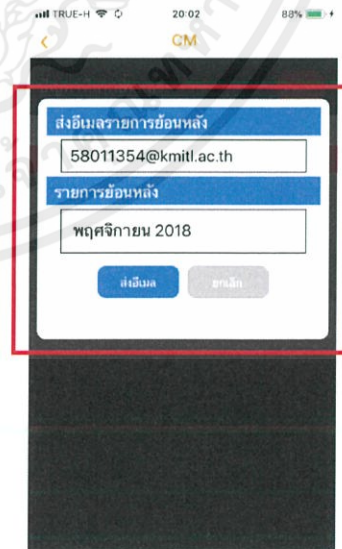
(ข)

รูปที่ 4.43 หน้าต่างแสดงปัญหาที่เกิดขึ้นแล้ว (ก) Android และ (ข) iOS

นอกจากนี้สามารถนำข้อมูลปัญหาที่เกิดขึ้นจัดทำเป็นไฟล์ข้อมูล Excel โดยทำการส่งอีเมล เริ่มต้นจากกดปุ่ม “ส่งออก” ดังรูปที่ 4.43 จะพามาหน้าต่างการส่งข้อมูลผ่านอีเมล จากนั้นทำการกรอกอีเมลและเลือกเดือนที่ต้องการข้อมูลตามรูปที่ 4.44 กดปุ่มส่งอีเมล เมื่อสำเร็จจะขึ้นการแจ้งเตือนว่า “Send email successfully” ดังรูปที่ 4.45 ไฟล์จะถูกส่งมาที่อีเมลดังรูปที่ 4.46 โดยไฟล์ข้อมูลจะเป็นไฟล์ Excel ไฟล์ข้อมูลจะเป็นไฟล์ที่แสดงข้อมูลปัญหาที่เคยเกิดขึ้นในเดือนนั้น ๆ แสดงในรูป 4.47 โดยในการทดสอบเลือกเดือนพฤษภาคม

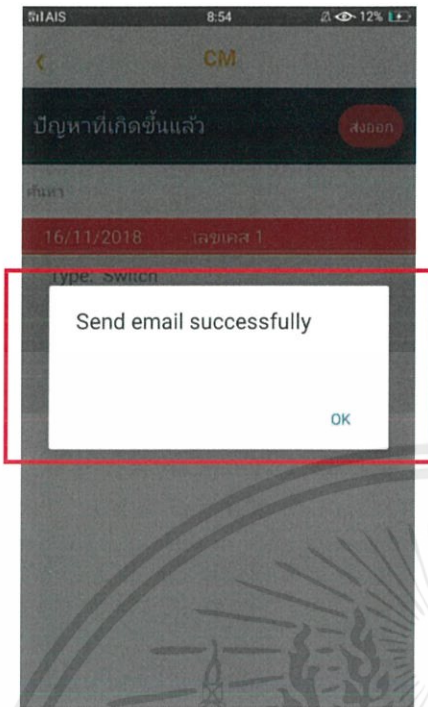


(ก)

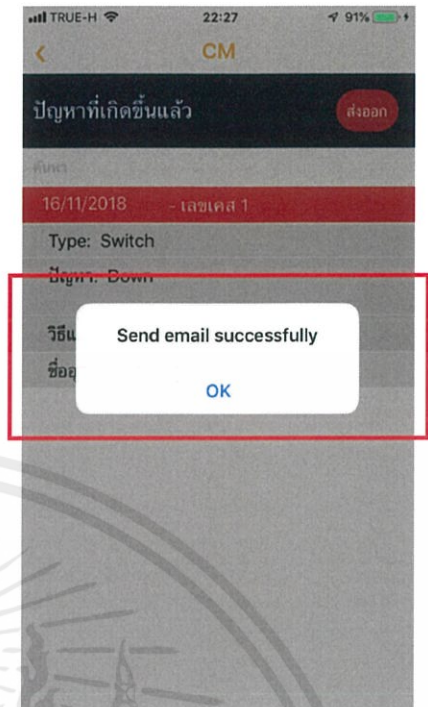


(ข)

รูปที่ 4.44 หน้าต่างกรอกอีเมลและเลือกเดือน (ก) Android และ (ข) iOS

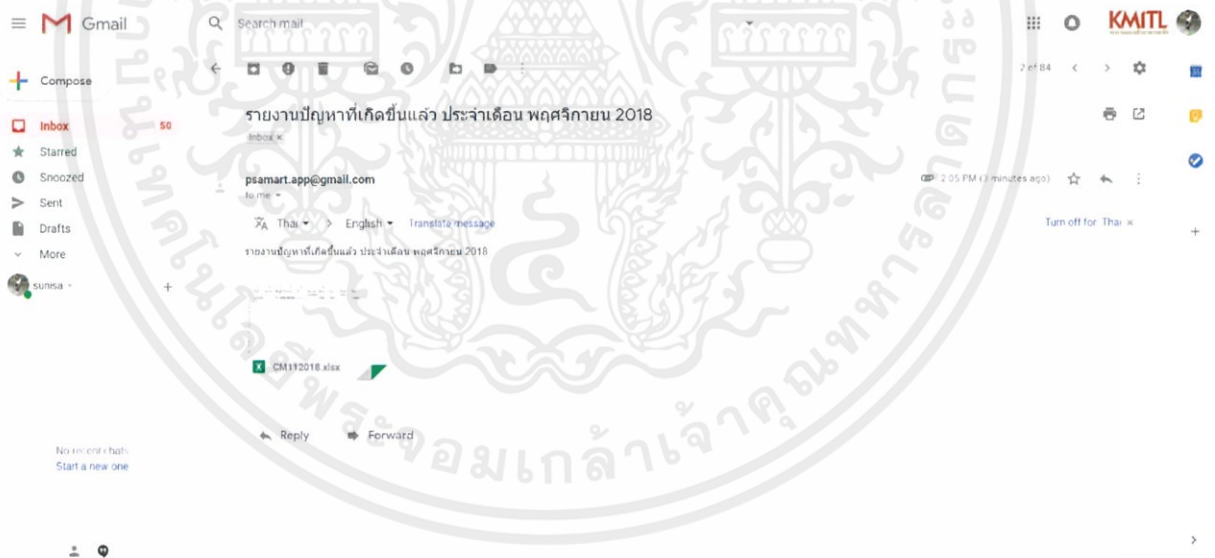


(ก)

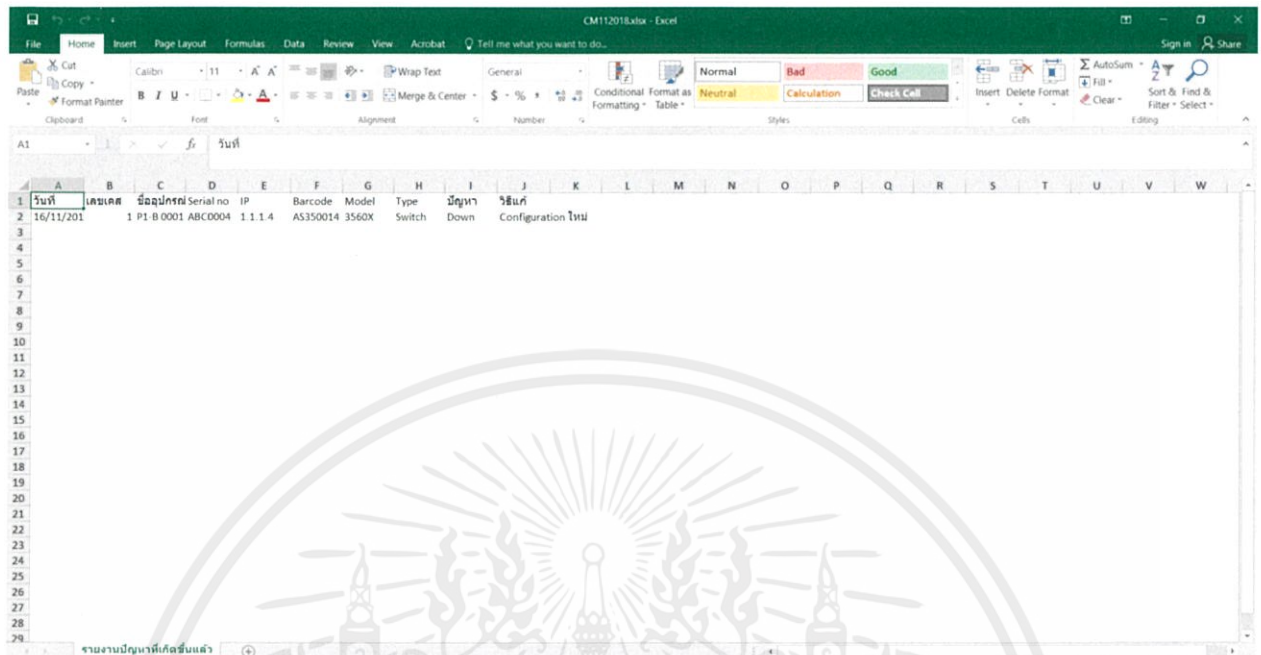


(ข)

รูปที่ 4.45 หน้าต่างแจ้งเตือนเมื่อส่งอีเมลสำเร็จ (ก) Android และ (ข) iOS



รูปที่ 4.46 หน้าต่างแสดงข้อความรายงานปัญหาที่เกิดขึ้นแล้วของงาน CM



รูปที่ 4.47 หน้าต่างโปรแกรม Excel แสดงรายงานปัญหาที่เกิดขึ้นแล้วของงาน CM

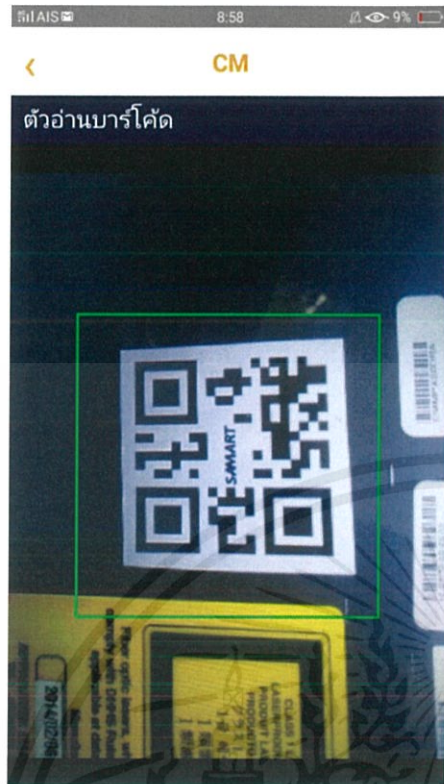
กรณีที่มีปัญหาเกิดขึ้นกับอุปกรณ์และมีการเปลี่ยนอุปกรณ์เกิดขึ้น ให้กดไปที่ปุ่มกด “แจ้งเปลี่ยนอุปกรณ์” ดังรูปที่ 4.48 จะนำไปหน้าต่างการแจ้งเปลี่ยนอุปกรณ์ดังรูปที่ 4.49 โดยทำการนำอุปกรณ์ออกจากคลังพัสดุก่อน จากนั้นทำการสแกนคิวอาร์โค้ดของอุปกรณ์เดิมและอุปกรณ์ใหม่ โดยกดปุ่มตัวสแกนคิวอาร์โค้ด ดังรูปที่ 4.49 จะนำไปหน้าต่างการสแกนคิวอาร์โค้ด โดยที่รูป 4.50 จะเป็นรูปการสแกนคิวอาร์โค้ดอุปกรณ์เดิมและรูปที่ 4.51 เป็นอุปกรณ์ใหม่ หลังจากสแกนเสร็จและข้อมูลจะขึ้นแสดงดังรูปที่ 4.52



รูปที่ 4.48 หน้าต่างหลักของ Corrective Maintenance (ก) Android และ (ข) iOS



รูปที่ 4.49 หน้าต่างการแจ้งเปลี่ยนอุปกรณ์ (ก) Android และ (ข) iOS



(ก)



(ข)

รูปที่ 4.50 หน้าต่างสแกนคิวอาร์โค้ดอุปกรณ์เดิม (ก) Android และ (ข) iOS



(ก)



(ข)

รูปที่ 4.51 หน้าต่างสแกนคิวอาร์โค้ดอุปกรณ์ใหม่ (ก) Android และ (ข) iOS

หน้าจอแสดงการแจ้งเปลี่ยนอุปกรณ์ (ก) บนระบบ Android

CM

แจ้งเปลี่ยนอุปกรณ์

อุปกรณ์เดิม

ชื่ออุปกรณ์: P1-B 0001

Model: 3560X

Serial No: ABC0004

Barcode No: AS350014

อุปกรณ์ใหม่

Model: C3560X

Serial No: ABC0005

Barcode No: AS350015

บันทึก ยกเลิก

หน้าจอแสดงการแจ้งเปลี่ยนอุปกรณ์ (ข) บนระบบ iOS

CM

แจ้งเปลี่ยนอุปกรณ์

อุปกรณ์เดิม

ชื่ออุปกรณ์: P1-B 0001

Model: C3560X

Serial No: ABC0004

Barcode No: AS350014

อุปกรณ์ใหม่

Model: C3560X

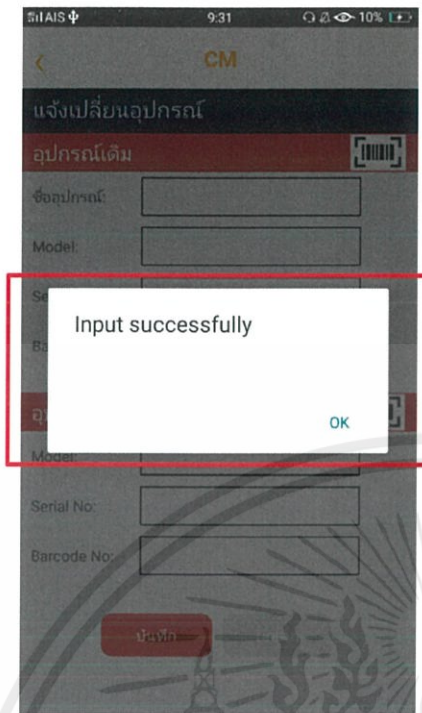
Serial No: ABC0005

Barcode No: AS350015

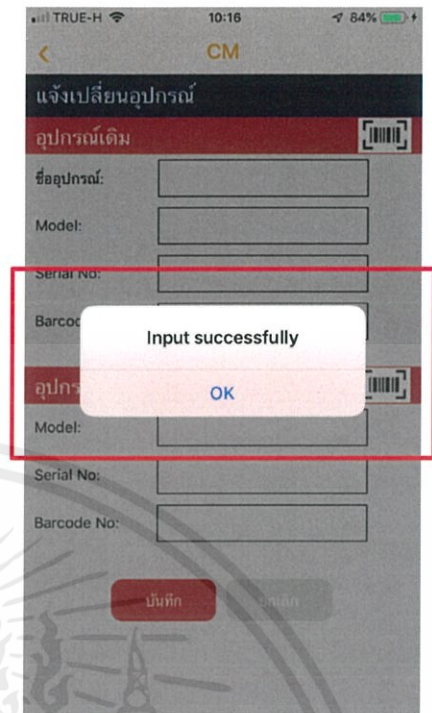
บันทึก ยกเลิก

รูปที่ 4.52 หน้าตาการแจ้งเปลี่ยนอุปกรณ์เมื่อสแกน QR Code (ก) Android และ (ข) iOS

จากนั้นทำการกดบันทึกจะขึ้นการแจ้งเตือนว่า “Input successfully” ดังแสดงในรูปที่ 4.53 และเมื่อกลับไปดูที่สถาน P1-B ซึ่งถูกทำการเปลี่ยนอุปกรณ์จะถูกเปลี่ยนเป็นชื่ออุปกรณ์ ข้อมูลต่างๆ เป็นของอุปกรณ์ใหม่แทนที่ทันทีดังรูปที่ 4.54 เปรียบเทียบก่อนเปลี่ยนอุปกรณ์และหลังเปลี่ยนอุปกรณ์และรูปที่ 4.55 เป็นรูปที่แสดงข้อมูลฐานข้อมูลของอุปกรณ์ก่อนเปลี่ยนอุปกรณ์ ในรูปที่ 4.56 เป็นรูปที่แสดงข้อมูลฐานข้อมูลของอุปกรณ์หลังเปลี่ยนอุปกรณ์ โดยจะมีการเปลี่ยนแปลงค่าข้อมูลเป็นของอุปกรณ์ใหม่



(ก)

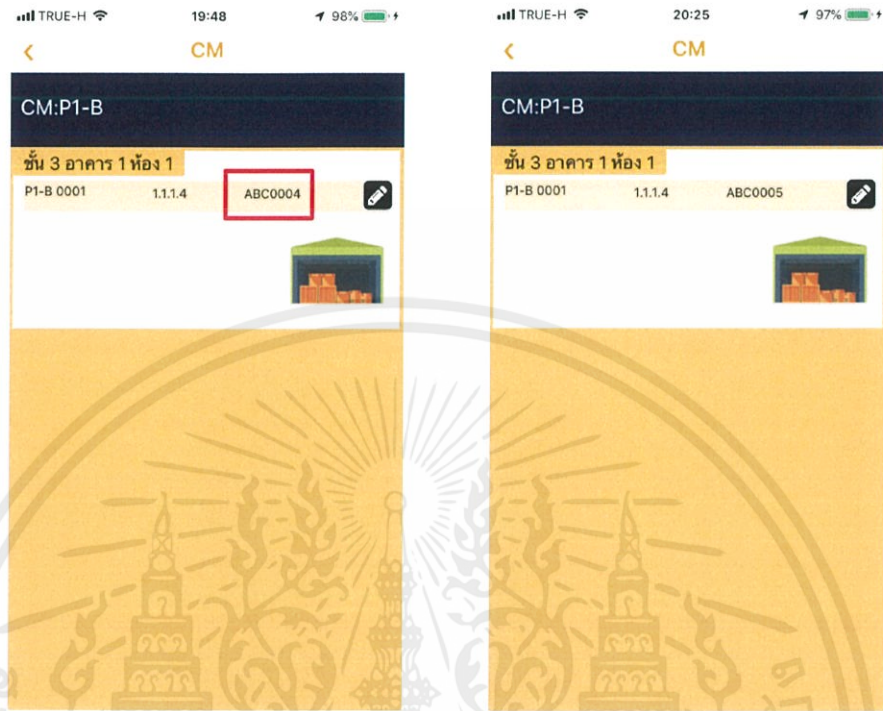


(ข)

รูปที่ 4.53 หน้าต่างการแจ้งเตือนเมื่อกดบันทึกการเปลี่ยนอุปกรณ์ (ก) Android และ (ข) iOS



(ก)



(ข)

รูปที่ 4.54 หน้าต่างข้อมูลเพิ่มเติมเปรียบเทียบเมื่อเปลี่ยนอุปกรณ์ (ก) Android และ (ข) iOS

device_id	place_1	building_id	room_id	device_name	device_no	device_type	device_ip	device_model	device_barcode	device_status	createby
1	P1-A	1	1	P1-A 0001	ABC0001	switch	1.1.1.1	3560X	AS350011	1	test
2	P1-A	1	1	P1-A 0002	ABC0002	switch	1.1.1.2	3560X	AS350012	1	test
3	P1-A	2	2	P1-A 0003	ABC0003	switch	1.1.1.3	3560X	AS350013	1	test
4	P1-B	3	3	P1-B 0001	ABC0004	switch	1.1.1.4	C3560X	AS350014	1	test

รูปที่ 4.55 ฐานข้อมูลตาราง “device” แสดงข้อมูลอุปกรณ์เดิม

device_id	place_1	building_id	room_id	device_name	device_no	device_type	device_ip	device_model	device_barcode	device_status	createby
1	P1-A	1	1	P1-A-0001	ABC0001	switch	1.1.1.1	3560X	AS350011	1	test
2	P1-A	1	1	P1-A-0002	ABC0002	switch	1.1.1.2	3560X	AS350012	1	test
3	P1-A	2	2	P1-A-0003	ABC0003	switch	1.1.1.3	3560X	AS350013	1	test
4	P1-B	3	3	P1-B-0001	ABC0005	switch	1.1.1.4	C3560X	AS350015	1	test

รูปที่ 4.56 ฐานข้อมูลตาราง “device” แสดงข้อมูลอุปกรณ์ใหม่

4.1.4 ทดสอบใช้งานแอปพลิเคชันของหน้าต่างคลังพัสดุ (Store)

หน้าต่างคลังพัสดุ เป็นหน้าต่างที่มีฟังก์ชันเกี่ยวกับการนำเข้า – นำออกของอุปกรณ์ภายในแผนก รวมทั้งสามารถตรวจสอบจำนวนอุปกรณ์แต่ละประเภทที่มีอยู่ในคลังพัสดุ ของแผนกได้ และสามารถดูย้อนหลังการนำเข้า – นำออกของอุปกรณ์ โดยเมื่อต้องการใช้ฟังก์ชันนี้ ให้ทำการ Login ให้สำเร็จ จากนั้นจะนำเรามาหน้า Home ให้เลือกกดปุ่ม “Store” ดังรูปที่ 4.57 หลังจากกดปุ่มจะนำเรามาหน้าต่างการฟังก์ชันการใช้งานดังรูปที่ 4.58 โดยมีฟังก์ชันการนำเข้าอุปกรณ์ การนำออกอุปกรณ์ คลังพัสดุ และรายละเอียดการนำเข้า – นำออกย้อนหลัง



รูปที่ 4.57 หน้าต่าง Home ของแอปพลิเคชัน (ก) Android และ (ข) iOS

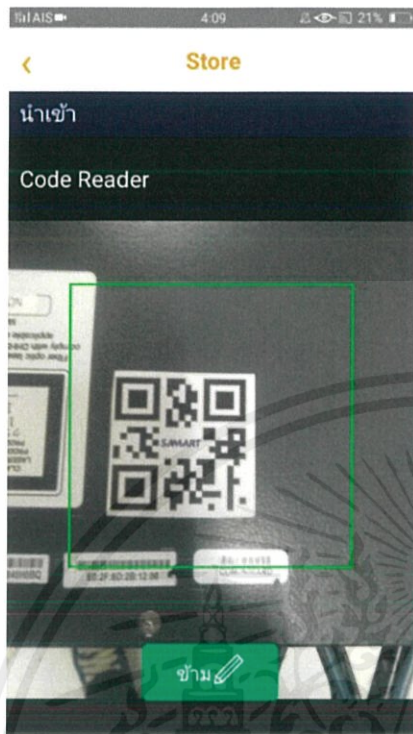


(ก)

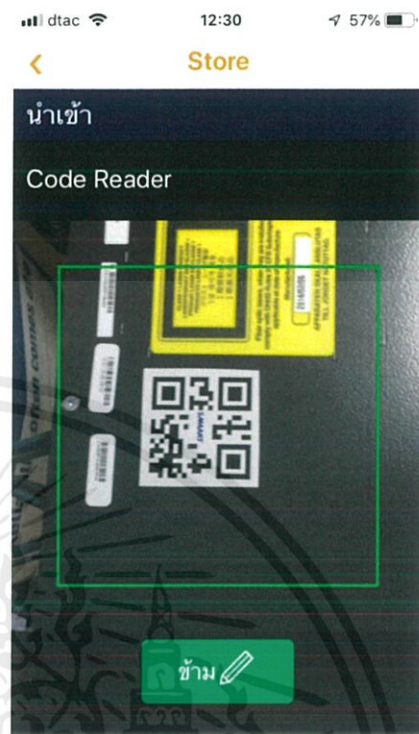
(ข)

รูปที่ 4.58 หน้าต่างหลักของ Store (ก) Android และ (ข) iOS

เมื่อเราต้องการนำอุปกรณ์เข้าคลังพัสดุให้กดปุ่ม “นำเข้า” ดังรูปที่ 4.58 จะพาไปหน้าต่างสแกน QR code ของการนำเข้าแสดงดังรูปที่ 4.59 และทำการสแกนดังรูปที่ 4.60 โดย QR code เป็นเลขครุภัณฑ์ของอุปกรณ์ ที่ผู้จัดทำเป็นผู้ออกแบบตามที่บริษัทต้องการ เมื่อทำการสแกน QR code ของอุปกรณ์สำเร็จจะพามาหน้าต่างกรอกข้อมูลของอุปกรณ์ดังรูปที่ 4.61 ทำการกรอกข้อมูล วันที่ Serial No. Model Type หมายเลข และผู้นำเข้า จากนั้นกดปุ่ม “บันทึก” จะขึ้นแจ้งเตือนว่า “Input successfully” ดังรูปที่ 4.62 และข้อมูลจะถูกบันทึกในฐานข้อมูลดังรูปที่ 4.63



(ก)

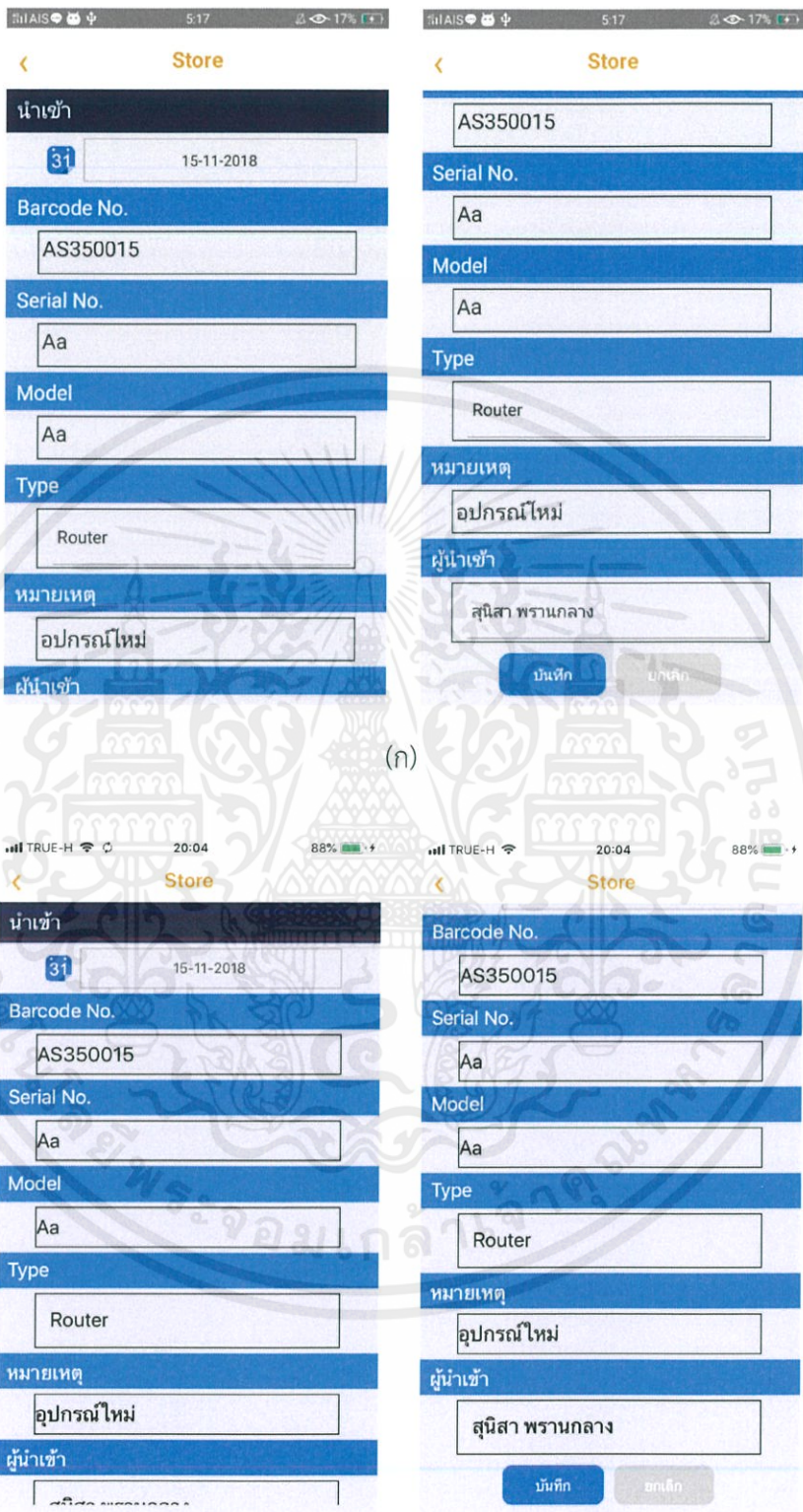


(ข)

รูปที่ 4.59 หน้าต่างอ่าน QR Code การนำเข้าอุปกรณ์ (ก) Android และ (ข) iOS



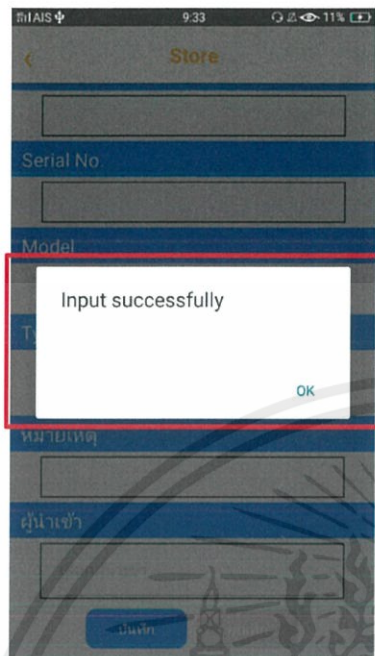
รูปที่ 4.60 แสดงการสแกน QR Code เพื่อนำเข้าอุปกรณ์



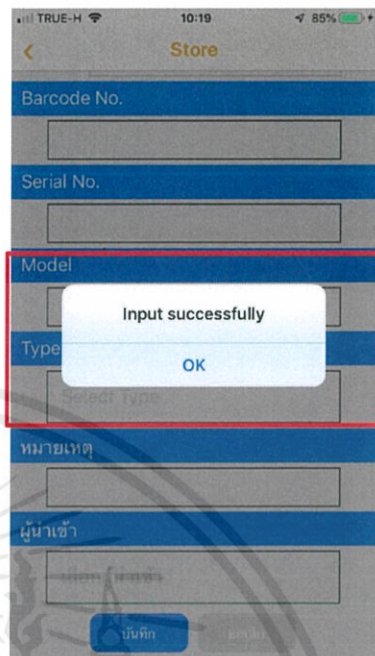
(ก)

(ข)

รูปที่ 4.61 หน้าต่างการนำเข้าอุปกรณ์แสดงข้อมูลการสแกน QR Code และกรอกข้อมูลเพื่อนำเข้าอุปกรณ์ (ก) Android และ (ข) iOS



(ก)



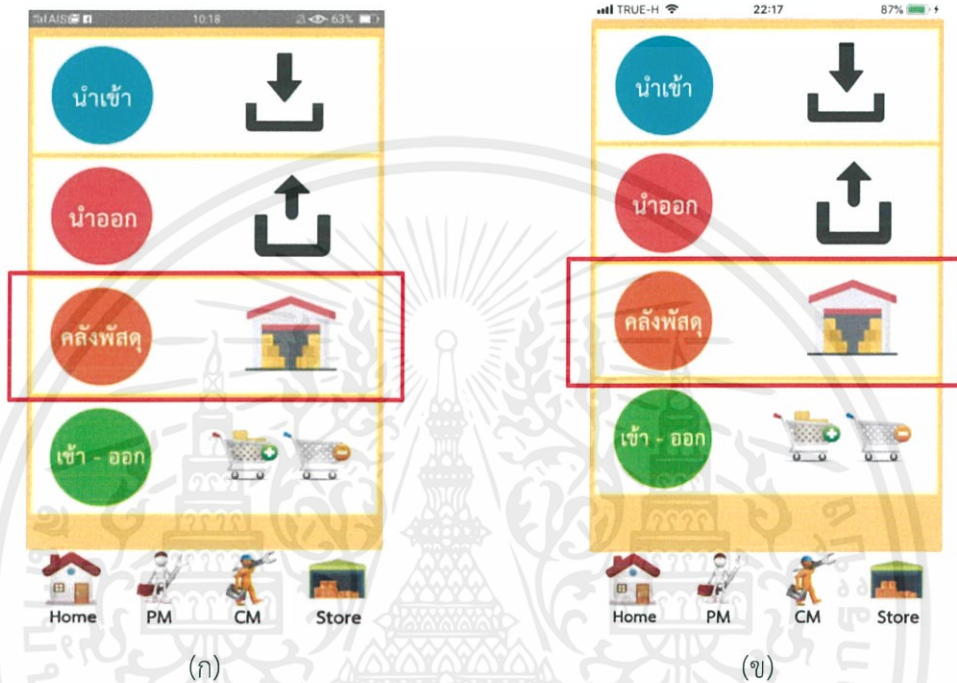
(ข)

รูปที่ 4.62 หน้าต่างแจ้งเตือนเมื่อกด “บันทึก” ข้อมูลนำเข้าอุปกรณ์ (ก) Android และ (ข) iOS

store_id	place_1	device_no	device_model	device_type	device_barcode	store_status	store_remark	store_by	store_date	createby	createdate
1	NULL	ABC0010	3560X	Switch	AS350020	1	NULL	สตีลา ทวีธาภิรมย์	2018-11-14	Test	2018-11-14 10:00:14
2	NULL	Aa	Aa	Router	AS350015	1	สตีลาทวีธาภิรมย์	สตีลา ทวีธาภิรมย์	2018-11-15	Test	2018-11-15 11:17:09

รูปที่ 4.63 ฐานข้อมูลตาราง “store”

จากนั้นกลับมาที่หน้าต่างหลักของคลังพัสดุ เพื่อกดเข้าไปดูที่คลังพัสดุดังรูปที่ 4.64 เพื่อทำการตรวจสอบว่าพัสดุที่ได้ทำการนำเข้าได้แสดงผลที่หน้าคลังพัสดุ โดยแสดงผลตามรูปที่ 4.65 เนื่องจากเราทำการนำเข้าพัสดุประเภท Router จะเห็นในรูปขึ้นคำมี Router 1 ตัว เมื่อทำการกดเข้าไปจะเห็นว่ามีรายการแสดงรายละเอียดอุปกรณ์ โดยแสดงข้อมูล Model และ Serial Number ของอุปกรณ์ตัวนั้นแสดงดังรูปที่ 4.66

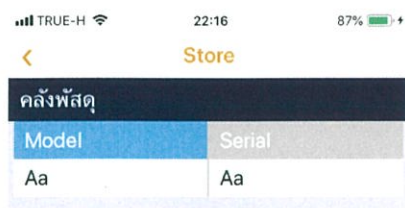


รูปที่ 4.64 หน้าต่างหลักของ Store (ก) Android และ (ข) iOS



รูปที่ 4.65 หน้าต่าง “คลังพัสดุ” (ก) Android และ (ข) iOS

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

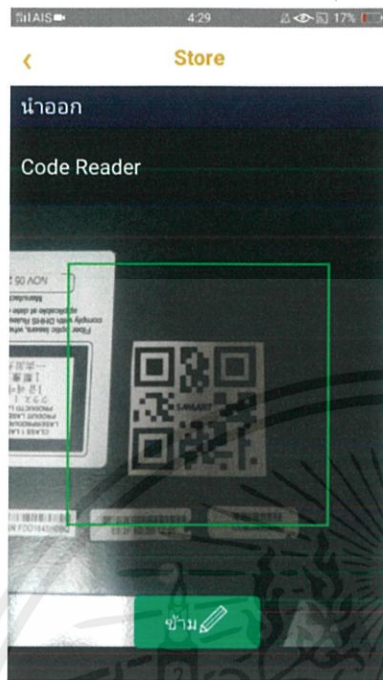


(ก)

(ข)

รูปที่ 4.66 หน้าต่างรายละเอียดอุปกรณ์ในคลังพัสดุ (ก) Android และ (ข) iOS

หลังจากนั้นทำการทดสอบฟังก์ชันการนำออก โดยกดปุ่ม “นำออก” ในหน้าเลือกต่างหลักของคลังพัสดุ จากนั้นจะนำมาหน้าสแกนคิวอาร์โค้ดของการนำออกตามรูปที่ 4.67 จากนั้นทำการสแกนดังรูปที่ 4.68 เมื่อเราสแกนสำเร็จจะพามาหน้าต่างกรอกข้อมูลการนำออก ในการสแกนข้อมูลของอุปกรณ์จะถูกแสดงผลในรูปที่ 4.69 ซึ่งกรอกเพิ่มเพียงสถานที่ที่นำไปใช้ หมายเหตุและผู้นำออก จากนั้นกด “บันทึก” จะแสดงข้อความแจ้งเตือนว่า “output successfully” แสดงในรูปที่ 4.70 และไปดูที่หน้าต่างคลังพัสดุ อุปกรณ์ประเภท Router จะไม่แสดงให้เห็นเพราะได้ทำการนำออกจากระบบเรียบร้อยแล้วดังรูปที่ 4.71

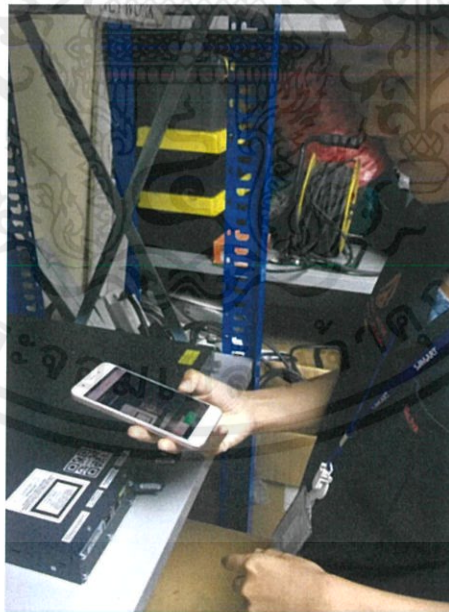


(ก)

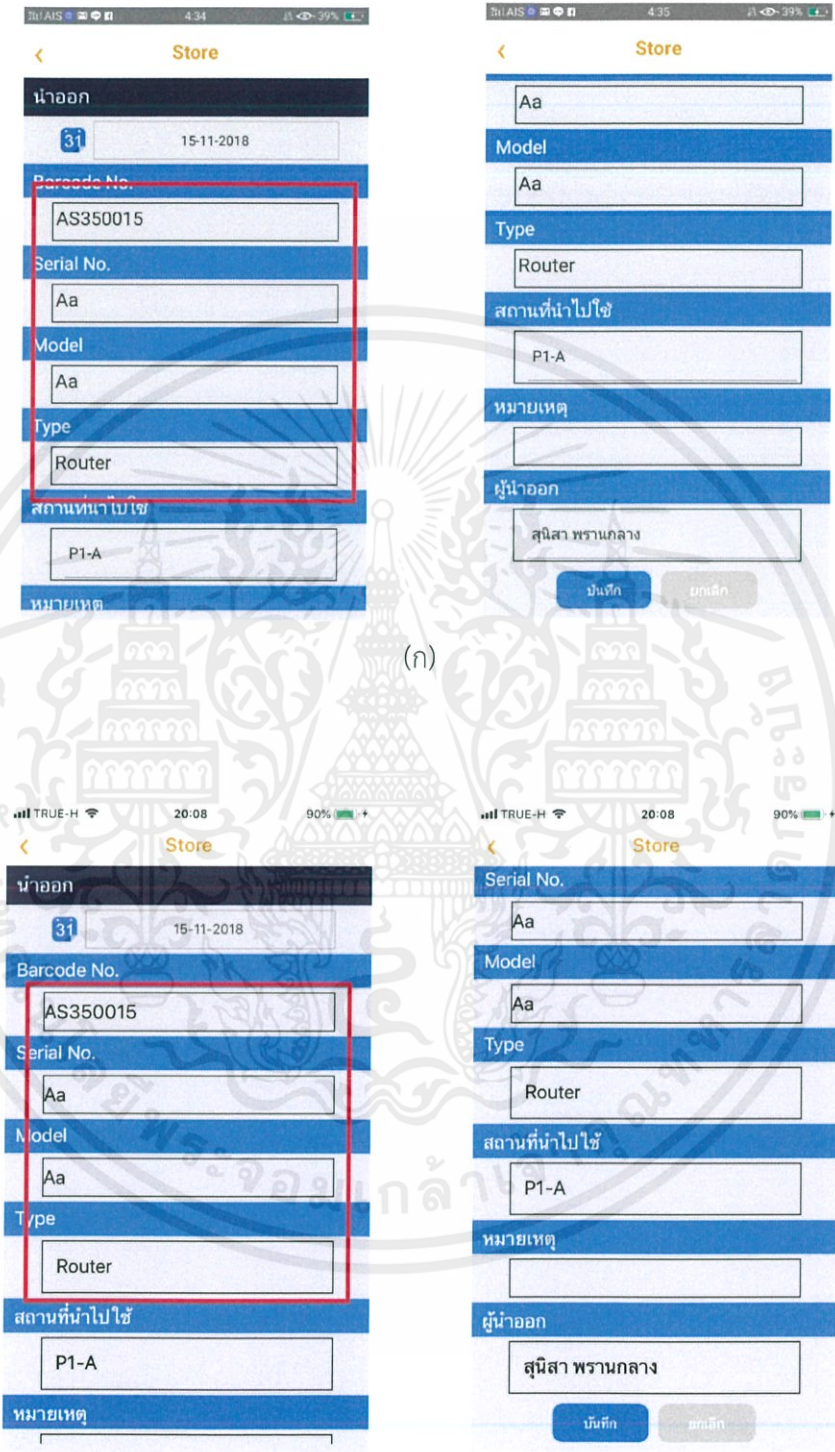


(ข)

รูปที่ 4.67 หน้าต่างอ่าน QR Code การนำออกอุปกรณ์ (ก) Android และ (ข) iOS



รูปที่ 4.68 แสดงการสแกน QR Code เพื่อนำออกอุปกรณ์



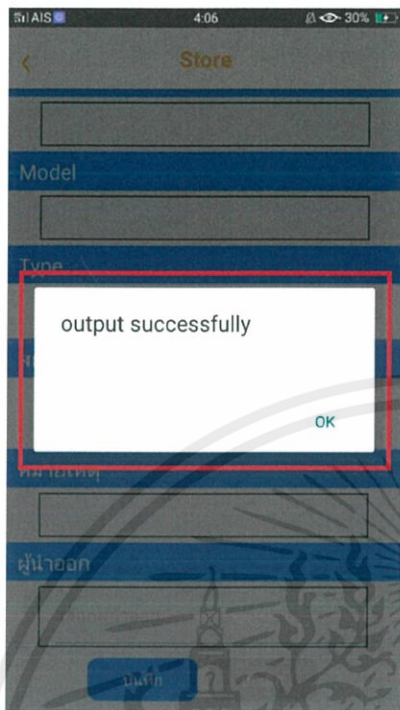
(ก)

(จ)

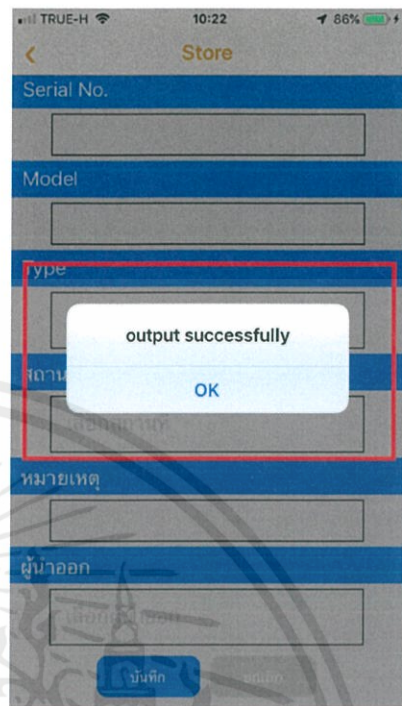
รูปที่ 4.69 หน้าต่างการนำออกอุปกรณ์แสดงข้อมูลการสแกน QR Code และกรอกข้อมูล

เพื่อนำออกอุปกรณ์ (ก) Android และ (จ) iOS

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

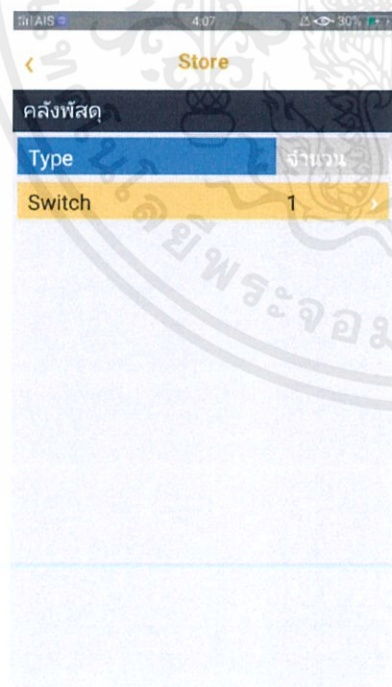


(ก)

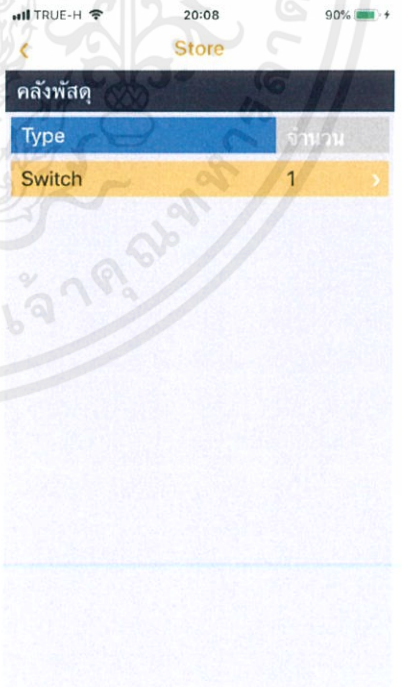


(ข)

รูปที่ 4.70 หน้าต่างแจ้งเตือนเมื่อกด “บันทึก” ข้อมูลนำออกอุปกรณ์ (ก) Android และ (ข) iOS



(ก)



(ข)

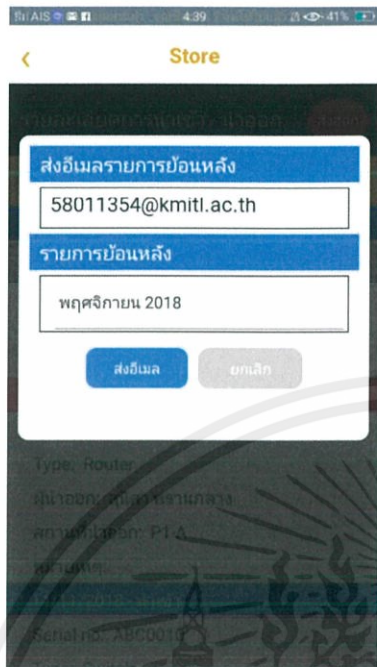
รูปที่ 4.71 หน้าต่าง “คลังพัสดุ” (ก) Android และ (ข) iOS

เมื่อต้องการดูรายละเอียดย้อนหลังการนำเข้า – นำออกพัสดุ สามารถทำได้โดยกดปุ่ม “เข้า – ออก” ที่หน้าต่างหลักของคลังพัสดุจะนำไปหน้าต่างที่แสดง “รายละเอียดการนำเข้า – นำออก” ของอุปกรณ์ต่างๆ และสามารถกดค้นหาได้ตามดังรูปที่ 4.72

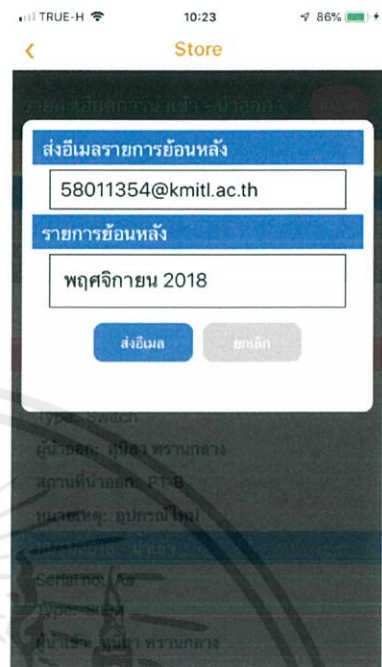


รูปที่ 4.72 หน้าต่าง “รายละเอียดการนำเข้า – ออก” (ก) Android และ (ข) iOS

นอกจากนี้สามารถนำข้อมูลการนำเข้า – นำออกของอุปกรณ์ และอุปกรณ์ที่อยู่ในคลังพัสดุดูออกมาเป็นไฟล์ Excel ได้โดยการส่งอีเมล เริ่มต้นจากกดปุ่ม “ส่งออก” ดังรูปที่ 4.72 จะพามาหน้าต่างการส่งข้อมูลผ่านอีเมล จากนั้นทำการกรอกอีเมลและเลือกเดือนที่ต้องการข้อมูลตามรูปที่ 4.73 กดปุ่ม “ส่งอีเมล” เมื่อสำเร็จจะขึ้นการแจ้งเตือนว่า “Send email successfully” ดังรูปที่ 4.74 ไฟล์จะถูกส่งมาที่อีเมลดังรูปที่ 4.75 โดยไฟล์ข้อมูลจะเป็นไฟล์ที่แสดงอุปกรณ์ของคลังพัสดุของแผนกในเดือนนั้นดังรูปที่ 4.76 และรายละเอียดการนำเข้า – ออกของอุปกรณ์ในเดือนนั้น ๆ ที่เลือก ดังรูปที่ 4.77 ในการทดสอบได้เลือกข้อมูลเดือนพฤศจิกายน ซึ่งในไฟล์จะมีรายการนำเข้า – ออก 3 รายการ ตรงตามที่ได้แสดงในหน้าต่างการนำเข้า – นำออกของอุปกรณ์ ในรูปที่ 4.72

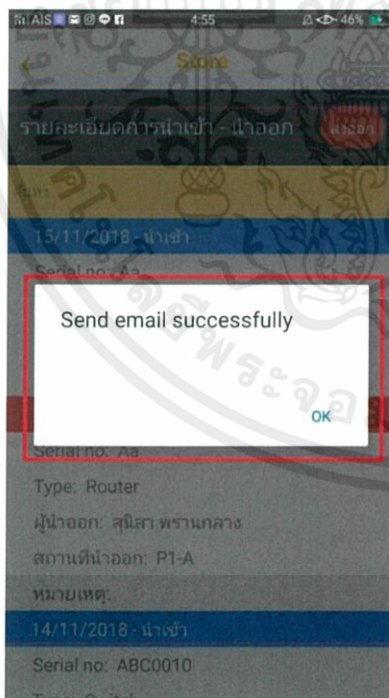


(ก)

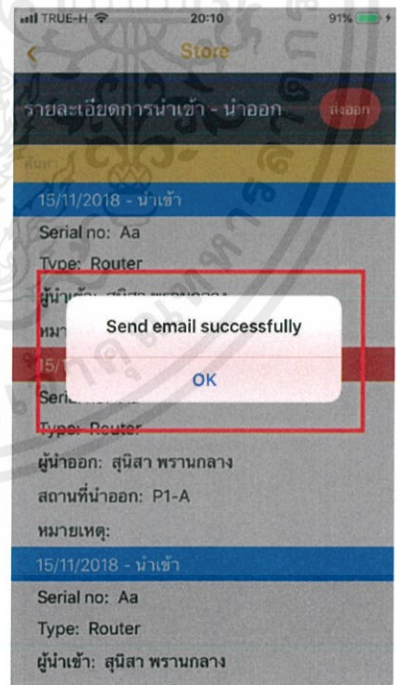


(ข)

รูปที่ 4.73 หน้าต่างกรอกอีเมลและเลือกเดือน (ก) Android และ (ข) iOS

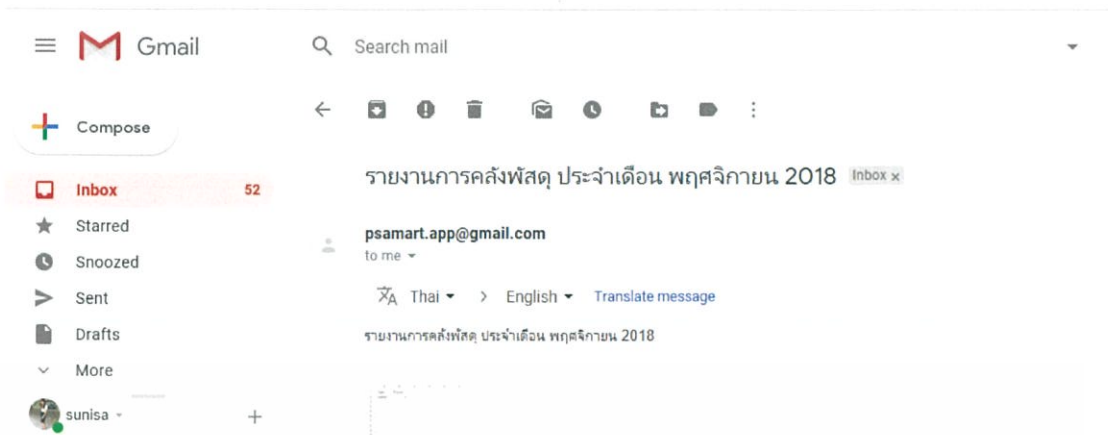


(ก)

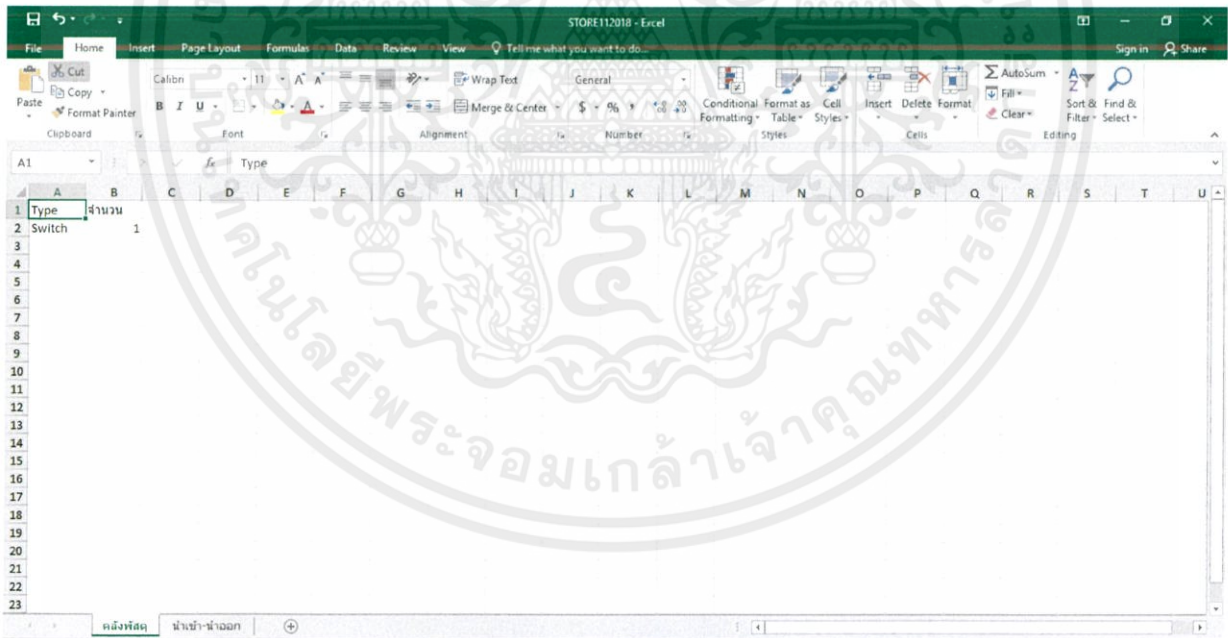


(ข)

รูปที่ 4.74 หน้าต่างแจ้งเตือนเมื่อส่งอีเมลสำเร็จ (ก) Android และ (ข) iOS



รูปที่ 4.75 หน้าต่างแสดงข้อความรายการคลังพัสดุ



รูปที่ 4.76 หน้าต่างโปรแกรม Excel แสดงรายงานคลังพัสดุ (จำนวน)

วันที่	Serial no	Barcode	Model	Type	สถานที่	สถานะ	ผู้เข้า/ออก	หมายเหตุ
15/11/201	Aa	AS350015	Aa	Router	PI-A	นำออก	ศูนย์ฯ พรานกลาง	อุปกรณ์ใหม่
15/11/201	Aa	AS350015	Aa	Router		นำเข้า	ศูนย์ฯ พรานกลาง	
14/11/201	ABC0010	AS350020	3560X	Switch		นำเข้า	ศูนย์ฯ พรานกลาง	

รูปที่ 4.77 หน้าต่างโปรแกรม Excel แสดงรายละเอียดการนำเข้าและนำออกของอุปกรณ์ในคลังพัสดุ

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการดำเนินงานทำโครงการ “ระบบบริหารจัดการซ่อมบำรุงรักษาเครือข่ายในสนามบิน” ได้ออกแบบและสร้างเป็นแอปพลิเคชันบนโทรศัพท์มือถือ สำหรับระบบปฏิบัติการแอนดรอยด์ (Android) และระบบปฏิบัติการไอโอเอส (iOS) เพื่อสะดวกต่อการใช้งาน จากการทดสอบการใช้งานแอปพลิเคชันสำหรับระบบปฏิบัติการแอนดรอยด์ (Android) และจำลองเหตุการณ์ในการใช้งานทั้งระบบ Login และ 3 ระบบการใช้งาน ได้แก่ ระบบการทำงานการบำรุงรักษาเชิงป้องกัน (Preventive Maintenance) ระบบการทำงานการบำรุงรักษาเชิงแก้ไข (Corrective Maintenance) และระบบคลังพัสดุ (Store)

ผลการทดสอบการใช้งานแอปพลิเคชันพบว่า แอปพลิเคชันสามารถใช้งานได้ทั้งระบบ Login และการทำงานของบริษัททั้ง 3 ระบบ โดยทำให้การทำงานของพนักงานสะดวกและรวดเร็วยิ่งขึ้น

1) ระบบการทำงานการบำรุงรักษาเชิงป้องกัน (Preventive Maintenance) สำหรับผู้ดูแลงาน สามารถเพิ่มงานให้พนักงานเข้าปฏิบัติงานตามสถานที่ต่าง ๆ ได้ และ สำหรับผู้ปฏิบัติงาน ระบบสามารถนำทางไปยังสถานที่ที่ปฏิบัติงานได้ แต่ต้องมีพิกัดสถานที่ดังกล่าวอยู่ในฐานข้อมูลด้วย สามารถกรอกข้อมูลแบบฟอร์มการบำรุงรักษาเครือข่ายได้ตามที่บริษัทกำหนด โดยระบบสามารถนำฐานข้อมูลที่กรอกจากแบบฟอร์มดังกล่าวออกมาเป็นไฟล์ Excel เพื่อใช้งานต่อได้ โดยการส่งอีเมล และระบบสามารถแจ้งสถานะการทำงานว่าสถานที่ดังกล่าวทำงานไปแล้วหรือไม่ โดยการระบุสี

2) ระบบการทำงานการบำรุงรักษาเชิงแก้ไข (Corrective Maintenance) ระบบนี้เมื่อมีปัญหาเกิดขึ้นกับตัวอุปกรณ์เครือข่าย ผู้ปฏิบัติงานจะต้องรีบดำเนินการแก้ไขปัญหาที่เกิดขึ้น ซึ่งระบบนี้สามารถค้นหาเส้นทางการเดินทางไปแก้ไขปัญหาอุปกรณ์ได้ เช่นเดียวกับระบบการทำงาน Preventive Maintenance สามารถกรอกแบบฟอร์มปัญหาที่เกิดขึ้นได้ ถ้าหากมีการเปลี่ยนอุปกรณ์เกิดขึ้นระบบนี้สามารถแจ้งเปลี่ยนอุปกรณ์ได้ โดยระบบสามารถอ่านคิวอาร์โค้ดมาเพื่อระบุตัวอุปกรณ์ และเมื่อกรอกแบบฟอร์มดังกล่าวแล้ว ในฟังก์ชัน “ปัญหาที่เกิดขึ้นแล้ว” จะนำข้อมูลจากฐานข้อมูลที่เคยกรอกในแบบฟอร์มปัญหาที่เคยเกิดขึ้นมาแสดงผลได้ ส่วนการแจ้งเปลี่ยนอุปกรณ์ เมื่อเปลี่ยนอุปกรณ์แล้ว ทำให้ข้อมูลอุปกรณ์ ณ สถานที่ต่าง ๆ เปลี่ยนการแสดงผลเป็น

อุปกรณ์ตัวใหม่ได้ และระบบสามารถนำฐานข้อมูลของปัญหาที่เคยเกิดขึ้นแล้วออกมาเป็นไฟล์ Excel เพื่อใช้งานต่อได้ โดยการส่งอีเมล

3) ระบบคลังพัสดุ (Store) การนำเข้าและการนำออกอุปกรณ์ได้ใช้เทคโนโลยีการอ่านคิวอาร์โค้ดมาช่วยในการระบุตัวอุปกรณ์ ซึ่งในระบบนี้สามารถใช้งานได้ หลังจากนำเข้าและนำออกอุปกรณ์ได้นำฐานข้อมูลดังกล่าวไปแสดงผลในฟังก์ชัน “คลังพัสดุ” และ “รายละเอียดการนำเข้า – ออก” ระบบสามารถนำฐานข้อมูลของคลังพัสดุและรายละเอียดการนำเข้าและนำออกออกมาเป็นไฟล์ Excel เพื่อใช้งานต่อได้ โดยการส่งอีเมล

การอ่านคิวอาร์โค้ดเพื่อระบุตัวอุปกรณ์สามารถใช้งานได้ แต่ถ้าแสงสว่างไม่เพียงพอหรือระยะห่างไม่ได้ก็อาจจะมีผลทำให้ระบบไม่สามารถอ่านได้หรืออ่านได้ช้า

5.2 ประโยชน์ของโครงการ

- 1) ช่วยสนับสนุนการทำงานให้สะดวกและรวดเร็วขึ้นเมื่อต้องปฏิบัติงาน Preventive Maintenance และงาน Corrective Maintenance เพียงใช้แอปพลิเคชันบนโทรศัพท์มือถือ
- 2) ช่วยนำทางไปยังสถานที่ปฏิบัติงานหรือตำแหน่งของอุปกรณ์เครือข่าย
- 3) ช่วยให้สามารถดูปัญหาที่เคยเกิดขึ้นกับอุปกรณ์เครือข่ายได้สะดวก รวดเร็วมากขึ้น
- 4) สนับสนุนการทำงานในส่วนระบบคลังพัสดุ (Store) สามารถจัดการได้สะดวกมากขึ้นและสามารถดูจำนวนอุปกรณ์ในคลังพัสดุง่ายขึ้น

5.3 ปัญหาและอุปสรรค

- 1) เนื่องจากถ้าหากต้องการใช้แอปพลิเคชันระบบปฏิบัติการไอโอเอส (iOS) จะต้องเสียค่าใช้จ่าย ทางบริษัทจึงให้ทดลองใช้เพียงแอปพลิเคชันระบบปฏิบัติการแอนดรอยด์ (Android)
- 2) เนื่องจากทางบริษัทไม่อนุญาตให้ใช้ฐานข้อมูลจริงและสถานการณ์จริงบางสถานการณ์ในการทดลองใช้งาน จึงทำให้ไม่สามารถได้เข้าไปดำเนินการในระบบการทำงานจริงได้อย่างเป็นทางการ

5.4 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

- 1) ควรพัฒนาและสร้างส่วนต่อประสานกับผู้ดูแลและจัดการฐานข้อมูล เพื่อสะดวกต่อการใช้งานของผู้ดูแลมากขึ้น
- 2) ควรพัฒนาและสร้างส่วนต่อประสานกับผู้ใช้งาน (User Interface) ในส่วนการลงทะเบียนเป็นผู้ใช้งานระบบ โดยต้องได้รับการยืนยันจากผู้ดูแลระบบ (Admin)
- 3) ควรพัฒนาให้ระบบสามารถนำออกข้อมูลจากฐานข้อมูลได้เป็นไฟล์ที่มีรูปแบบฟอร์มตามที่บริษัทใช้งาน ซึ่งทำให้สามารถใช้งานได้เลย



บรรณานุกรม

- [1] Kittisak Pimnonthong. “ทำความรู้จักกับ React Native ก่อนเริ่มต้นใช้งาน.”
https://medium.com/@dreamtery/ทำความรู้จักกับ-react-native-ก่อนเริ่มต้นใช้งาน-a0a9fe6da4d?fbclid=IwAR26GmbI8ktzhkRwsjmmSwdXbspUVOSQO_oMwy5wQgeRXBlxwH9-cg0UY9w.
- [2] MindPHP.com. “JavaScript คืออะไร.”
<https://www.mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html?fbclid=IwAR2sVKCpplx9mUkytL5KnU16DqT9ZTMg5ijGUw8qVlhZj61uEchfP2e4e2o>.
- [3] MindPHP.com. “รู้จักกับ Visual Studio Code (วิชวล สตูดิโอ โค้ด).”
https://mindphp.com/บทความ/microsoft/4829-visual-studio-code.html?fbclid=IwAR2m-VFBAUppbbf5Cn0B1ZT8wiVJSs0hEqLa1-9U8P8kb_ZMC9R883jr538.
- [4] Anuchit Chalothorn. “Genymotion Android Emulator.”
<http://thaiopensource.org/มาใช้-genymotion-android-emulator-กัน/>.
- [5] Banyapon Poolsawasd. “รู้จักกับ Xcode IDE.”
<http://www.daydev.com/developer/s6-programming-language/ios-develop-chapter-1-xcode.html>.
- [6] WordPress.com. “ระบบปฏิบัติการ ANDROID.”
<https://beerkung.wordpress.com/ระบบปฏิบัติการรุ่นล่าสุด/ระบบปฏิบัติการ-android/?fbclid=IwAR2dl6-2Uk4lcf2dtMMGxL0EDnh0MaFOBQO3uNfueL3ryxqHoq8Z3htJc7c>.
- [7] WordPress.com. “ระบบปฏิบัติการ IOS.”
<https://beerkung.wordpress.com/ระบบปฏิบัติการรุ่นล่าสุด/ระบบปฏิบัติการ-ios/?fbclid=IwAR0VPhwYVSJQ5NvsrTPre5CSS5peks7fjhnp-pMWwv2Y1NPlmQWPEpBAfhk>.

- [8] Easy Branches. “MySQL มีความสำคัญอย่างไรกับเซิร์ฟเวอร์.”
http://th.easyhostdomain.com/dedicated-servers/mysql.html?fbclid=IwAR3kLeQVcCDKU3W8-SexJlcz4_Btaq5DjxDFkykNpJctnV3moPSaZif_Kl.
- [9] MindPHP.com. “Xampp คืออะไร.”
https://mindphp.com/คู่มือ/73-คืออะไร/2637-xampp-คืออะไร.html?fbclid=IwAR3kLeQVcCDKU3W8-SexJlcz4_Btaq5DjxDFkykNpJctnV3moPSaZif_Kl.
- [10] MindPHP.com. “PHP คืออะไร.”
<https://www.mindphp.com/คู่มือ/73-คืออะไร/2127-php-คืออะไร.html?fbclid=IwAR182oed2-Bg3tqukQLiLBQRdl7FCR14fzzE5OfLCzhvSc3TJN6jUM9zLXI>.
- [11] THAICODING. “Marvel App เครื่องมือที่ช่วยออกแบบ UX/UI”
<https://thaicoding.net/2017/06/21/รีวิว-marvel-app-เครื่องมือที่ช่วย/>.
- [12] Riverplus. “Barcode คืออะไร.”
https://riverplusblog.com/2011/06/07/barcode-คืออะไร/?fbclid=IwAR2x_-2RwT_Nb8frYJvguBReuHlAiefWeMxfnKXtZv2lhOFxqZQG8icuVgY.
- [13] Telzel co.,ltd. “มาตรฐาน บาร์โค้ดประเภทต่าง ๆ.”
<https://www.telzel.com/know4.html>.
- [14] ThaiCreate.com “Google Maps API (JavaScript) เบื้องต้นกับการใช้งาน.”
<https://www.thaicreate.com/tutorial/google-maps-javascript-api.html>.
- [15] Swiftlet. “GOOGLE MAP API คืออะไร.”
https://swiftlet.co.th/google-api-คืออะไร/?fbclid=IwAR1PG8rVb04U-XKS w7EL_8ZAJwdLmehxkFin2Q8igbPHXnLrtawxTp6eUlq.
- [16] MakeWebEasy. “วิธีการขอ GOOGLE MAP API KEY สำหรับ GOOGLE MAP.”
https://www.makewebeasy.com/blog/2016/06/google-map-api-key-manual/?fbclid=IwAR2R_2x6nXRehIFjke53uhH7lV_KqRj4Qo4LL0TEKaltDfnHTye1DARyook.



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

api.js

```
const ipAddress = '1.1.1.1';
const endpoint = 'http://'+ipAddress+'/database_project/';
export const callApi = (method) => {
  return endpoint+method; }
export const getLastMonth = (offsize) => {
  offsize = (offsize==null || offsize=="") ? 12 : offsize;
  const monthNames = ["มกราคม","กุมภาพันธ์","มีนาคม","เมษายน","พฤษภาคม","มิถุนายน", "กรกฎาคม","สิงหาคม","กันยายน","ตุลาคม","พฤศจิกายน","ธันวาคม"];
  const day = new Date();
  let month = day.getMonth();
  let year = day.getFullYear();
  let months = [];
  for(let i = 0;i<offsize; i++) {
    let name = monthNames[month];
    months[i] = {};
    months[i].label = name+' '+year;
    months[i].value = (month+1)+' '+year;
    month = (month-1);
    if(month < 0) {
      month = 11;
      year = (year-1);
    }
  }
  return months;}
export default {
  USERNAME : "",
  old_device_name : "",
  old_device_model : "",
  old_device_no : "",
  old_device_barcode : "",
  new_device_model : "",
  new_device_no : "",
  new_device_barcode : ""};
```

Menu.js

```
import React, { Component } from 'react';
import { StyleSheet, View, Image,TouchableOpacity,
} from 'react-native';
export default class Menu extends Component {
  constructor(props) {
    super(props); }
  render() {
    const { navigation } = this.props;
    return (
      <View style={styles.menuListStyle}>
        <TouchableOpacity style={styles.menuitemStyle} onPress={() =>
navigation.replace('page2')}>
          <Image source={require('./b/1.jpg')} style={{ width: 90, height: 80, }} />
        </TouchableOpacity>
        <TouchableOpacity style={styles.menuitemStyle} onPress={() =>
navigation.replace('page3')}>
          <Image source={require('./b/2.jpg')} style={{ width: 90, height: 80, }} />
        </TouchableOpacity>
        <TouchableOpacity style={styles.menuitemStyle} onPress={() =>
navigation.replace('page7')}>
          <Image source={require('./b/3.jpg')} style={{ width: 90, height: 80, }} />
        </TouchableOpacity>
        <TouchableOpacity style={styles.menuitemStyle} onPress={() =>
navigation.replace('page17')}>
          <Image source={require('./b/4.jpg')} style={{ width: 90, height: 80, }} />
        </TouchableOpacity>
      </View>
    );
    const styles = StyleSheet.create({
      menuListStyle: {
        position: 'absolute',
        flexDirection: 'row',
        backgroundColor: '#FFF',
        bottom: 0,
        left: 0,
        width: '100%', },
    });
  }
}
```

```
menuItemStyle: {
  width: '25%',
  height: 80,
  padding: 1,
},
});
```

Navigation.js

```
import React from 'react';
import { View,TouchableOpacity,Text } from 'react-native';
import { createStackNavigator,StackActions } from 'react-navigation';
import page1 from './page1';
import page2 from './page2';
import page3 from './page3';
import page3_1 from './page3_1';
import page4 from './page4';
import page5 from './page5';
import page6 from './page6';
import page7 from './page7';
import page7_1 from './page7_1';
import page8 from './page8';
import page9 from './page9';
import page10 from './page10';
import page11 from './page11';
import page12 from './page12';
import page13 from './page13';
import page14 from './page14';
import page15 from './page15';
import page17 from './page17';
import page18 from './page18';
import page19 from './page19';
import page20 from './page20';
import page21 from './page21';
import page22 from './page22';
```

```

import page23 from './page23';
import page24 from './page24';
const arrow = "<";
const goReload = (navigation,page) => {
  const pushAction = StackActions.replace({routeName: page});
  navigation.dispatch(pushAction); };
const goBack = (navigation) => {
  navigation.pop();};
const goPop = (navigation,page) => {
  const pushAction = StackActions.pop({routeName: page});
  navigation.dispatch(pushAction) };
export const App = createStackNavigator({
  page1: {
    screen: page1,
    navigationOptions: ({ navigation }) => ({
      title: "",
      headerStyle: styles.headerHideStyle,
      headerRight: <View />,
      headerLeft: <View />,}),},
  page2: {
    screen: page2,
    navigationOptions: ({ navigation }) => ({
      title: "",
      headerStyle: styles.headerHideStyle,
      headerRight: <View />,
      headerLeft: <View />,}),},
  page3: {
    screen: page3,
    navigationOptions: ({ navigation }) => ({
      title: "",
      headerStyle: styles.headerHideStyle,
      headerRight: <View />,
      headerLeft: <View />,}),},
  page3_1: {
    screen: page3_1,
    navigationOptions: ({ navigation }) => ({

```

```

title: "PM",
headerTitleStyle: styles.headerTitleStyle,
headerStyle: styles.headerStyle,
headerRight: <View />,
headerLeft: (
  <TouchableOpacity onPress={() => goReload(navigation,'page3')}>
    <Text style={styles.headerLeftStyle}>{arrow}</Text>
  </TouchableOpacity>),)),),
page4: {
  screen: page4,
  navigationOptions: ({ navigation }) => ({
    title: "PM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goReload(navigation,'page3')}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>),)),),
page5: {
  screen: page5,
  navigationOptions: ({ navigation }) => ({
    title: "PM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goBack(navigation)}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>
page6: {
  screen: page6,
  navigationOptions: ({ navigation }) => ({
    title: "PM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,

```

```

headerRight: <View />,
headerLeft: (
  <TouchableOpacity onPress={() => goBack(navigation)}>
    <Text style={styles.headerLeftStyle}>{arrow}</Text>
  </TouchableOpacity>),), },
page7: {
  screen: page7,
  navigationOptions: ({ navigation }) => ({
    title: "",
    headerStyle: styles.headerHideStyle,
    headerRight: <View />,
    headerLeft: <View />,}),),
page7_1: {
  screen: page7_1,
  navigationOptions: ({ navigation }) => ({
    title: "CM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goReload(navigation,'page7')}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>),), },
page8: {
  screen: page8,
  navigationOptions: ({ navigation }) => ({
    title: "CM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goBack(navigation)}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>
page9: {
  screen: page9,

```

```

navigationOptions: ({ navigation }) => ({
  title: "CM",
  headerTitleStyle: styles.headerTitleStyle,
  headerStyle: styles.headerStyle,
  headerRight: <View />,
  headerLeft: (
    <TouchableOpacity onPress={() => goReload(navigation,'page8')}>
      <Text style={styles.headerLeftStyle}>{arrow}</Text>
    </TouchableOpacity>
  )
})

page10: {
  screen: page10,
  navigationOptions: ({ navigation }) => ({
    title: "CM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goBack(navigation)}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>
    )
  })
}

page11: {
  screen: page11,
  navigationOptions: ({ navigation }) => ({
    title: "CM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goBack(navigation)}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>
    )
  })
}

page12: {
  screen: page12,
  navigationOptions: ({ navigation }) => ({
    title: "CM",
    headerTitleStyle: styles.headerTitleStyle,
  })
}

```

```

headerStyle: styles.headerStyle,
headerRight: <View />,
headerLeft: (
  <TouchableOpacity onPress={() => goReload(navigation,'page7')}>
    <Text style={styles.headerLeftStyle}>{arrow}</Text>
  </TouchableOpacity>
)
page13: {
  screen: page13,
  navigationOptions: ({ navigation }) => ({
    title: "CM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goBack(navigation)}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>
    )
  })
}
page14: {
  screen: page14,
  navigationOptions: ({ navigation }) => ({
    title: "CM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goBack(navigation)}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>
    )
  })
}
page15: {
  screen: page15,
  navigationOptions: ({ navigation }) => ({
    title: "CM",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (

```

```

        <TouchableOpacity onPress={() => goBack(navigation)}>
            <Text style={styles.headerLeftStyle}>{arrow}</Text>
        </TouchableOpacity>)),),
page17: {
    screen: page17,
    navigationOptions: ({ navigation }) => ({
        title: "",
        headerStyle: styles.headerHideStyle,
        headerRight: <View />,
        headerLeft: <View />,
    }),
},
page18: {
    screen: page18,
    navigationOptions: ({ navigation }) => ({
        title: "Store",
        headerTitleStyle: styles.headerTitleStyle,
        headerStyle: styles.headerStyle,
        headerRight: <View />,
        headerLeft: (
            <TouchableOpacity onPress={() => goReload(navigation,'page17')}>
                <Text style={styles.headerLeftStyle}>{arrow}</Text>
            </TouchableOpacity>)),),
page19: {
    screen: page19,
    navigationOptions: ({ navigation }) => ({
        title: "Store",
        headerTitleStyle: styles.headerTitleStyle,
        headerStyle: styles.headerStyle,
        headerRight: <View />,
        headerLeft: (
            <TouchableOpacity onPress={() => goReload(navigation,'page18')}>
                <Text style={styles.headerLeftStyle}>{arrow}</Text>
            </TouchableOpacity>)),),
page20: {
    screen: page20,

```

```

navigationOptions: ({ navigation }) => ({
  title: "Store",
  headerTitleStyle: styles.headerTitleStyle,
  headerStyle: styles.headerStyle,
  headerRight: <View />,
  headerLeft: (
    <TouchableOpacity onPress={() => goReload(navigation,'page17')}>
      <Text style={styles.headerLeftStyle}>{arrow}</Text>
    </TouchableOpacity>
  )
})

page21: {
  screen: page21,
  navigationOptions: ({ navigation }) => ({
    title: "Store",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goReload(navigation,'page20')}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>
    )
  })
}

page22: {
  screen: page22,
  navigationOptions: ({ navigation }) => ({
    title: "Store",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goReload(navigation,'page17')}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>
    )
  })
}

page23: {
  screen: page23,
  navigationOptions: ({ navigation }) => ({
    title: "Store",
    headerTitleStyle: styles.headerTitleStyle,
  })
}

```

```

headerStyle: styles.headerStyle,
headerRight: <View />,
headerLeft: (
  <TouchableOpacity onPress={() => goBack(navigation)}>
    <Text style={styles.headerLeftStyle}>{arrow}</Text>
  </TouchableOpacity>
)
page24: {
  screen: page24,
  navigationOptions: ({ navigation }) => ({
    title: "Store",
    headerTitleStyle: styles.headerTitleStyle,
    headerStyle: styles.headerStyle,
    headerRight: <View />,
    headerLeft: (
      <TouchableOpacity onPress={() => goReload(navigation,'page17')}>
        <Text style={styles.headerLeftStyle}>{arrow}</Text>
      </TouchableOpacity>
    ),
  })
}, {
  initialRouteName: 'page1',
  mode: 'card'
});
const styles = {
  headerHideStyle: {
    height: 0,
  },
  headerTitleStyle: {
    color: '#e89b00',
    textAlign: 'center',
    fontSize: 20,
    flex: 1, },
  headerStyle : {
    backgroundColor: '#FFFFFF'},
  headerLeftStyle : {
    color:'#e89b00',

```

```
    fontSize:36,  
    marginLeft:10  
  }  
}
```

page1.js

```
import React, { Component } from 'react';  
import { callApi } from './api';  
import { Platform, StyleSheet, Text, View, Image, TextInput, TouchableOpacity, Alert, AsyncStorage  
} from 'react-native';  
import { StackActions } from 'react-navigation';  
export default class Login extends Component {  
  constructor(props) {  
    super(props)  
    this.state = {  
      user_name: "",  
      userpassword: ""  
    }  
  }  
  componentWillMount = async () => {  
    try {  
      const value = await AsyncStorage.getItem('user_name');  
      if (value != null) {  
        callApi.USERNAME = value;  
        const pushAction = StackActions.push({routeName: 'page2'});  
        this.props.navigation.dispatch(pushAction);  
      }  
    } catch (error) {  
    }  
  }  
  userlogin = () => {  
    const { user_name } = this.state;  
    const { userpassword } = this.state;  
    let login_url = callApi('login.php');  
    fetch(login_url, {  
      method: 'POST',  
      headers: {  
        'Accept': 'application/json',
```

```

    'Content-Type': 'application/json' },
    body: JSON.stringify({
      user: user_name,
      password: userpassword,
    })
  }).then((response) => response.json())
    .then((responseJson) => {
      if (responseJson.status) {
        AsyncStorage.setItem('user_name', user_name);
        this.setState({ user_name: "", userpassword: "" });
        const pushAction = StackActions.replace({routeName: 'page2'});
        this.props.navigation.dispatch(pushAction);
      } else {
        Alert.alert(responseJson.message);
        AsyncStorage.removeItem('user_name'); }
    }).catch((error) => {
      console.error(error);
      AsyncStorage.removeItem('user_name');
    });
  });
  render() {
    return (
      <View style={{ flex: 1, }}>
        <View style={{ flex: 1, backgroundColor: '#FFFF99', alignItems: 'center', }}>
          <View style={{ backgroundColor: '#FF9933', width: '100%', height: 120, alignItems:
'center', }}>
            <Image source={require('./a/header.jpg')} style={{ width: '80%', height: 100,
marginTop: 10, marginBottom: 0, }} />
          </View>
          <Image source={require('./a/icon_login.png')} style={{ width: '25%', height: '25%', }}
resizeMode='center' />
          <View style={{ backgroundColor: '#FFFFFF', width: 230 }}>
            <TextInput placeholder="Username"
              onChangeText={user_name => this.setState({ user_name })}
              style={{ width: 200, margin: 10 }}
              value={this.state.user_name}
              underlineColorAndroid='transparent'/>

```

```

</View>
<View style={{ backgroundColor: '#FFFFFF', marginTop: 15, width: 230 }}>
  <TextInput placeholder="Password"
    onChangeText={userpassword => this.setState({ userpassword })}
    secureTextEntry style={{ width: 200, margin: 10 }}
    value={this.state.userpassword}
    underlineColorAndroid='transparent'/>
</View>
<TouchableOpacity onPress={this.userlogin} style={{ width: 100, padding: 10,
backgroundColor: 'black', marginTop: 10, alignItems: 'center' }}>
  <Text style={{ color: 'white' }}>Login</Text>
</TouchableOpacity>
</View>
</View>

```

page2.js

```

import React, { Component } from 'react';
import { Platform, StyleSheet, Text, View, Image, Button, TouchableOpacity, ScrollView,
AsyncStorage
} from 'react-native';
import { StackActions } from 'react-navigation';
export default class page2 extends Component {
  btnLogout () {
    AsyncStorage.removeItem('user_name');
    const pushAction = StackActions.replace({routeName: 'page1'});
    this.props.navigation.dispatch(pushAction);
  }
  goPage (page) {
    const pushAction = StackActions.replace({routeName: page});
    this.props.navigation.dispatch(pushAction);
  };
  render() {
    return (
      <ScrollView style={styles.container}>
        <View style={{ flexDirection: 'row'}}>

```

```

<Image resizeMode='stretch' style={styles.SmartIMG}
  source={require('./img/w4.jpg')} />
<TouchableOpacity style={styles.Logoutbotton onPress={() => this.btnLogout()}>
  <Text style={{ color: '#FFFFFF', fontSize: 20 }}>Logout</Text>
</TouchableOpacity>
</View>
<TouchableOpacity onPress={() => this.goPage('page3')}>
  <Image resizeMode='contain' style={styles.BottonWork}
    source={require('./img/w1.jpg')} />
</TouchableOpacity>
<TouchableOpacity onPress={() => this.goPage('page7')}>
  <Image resizeMode='contain' style={styles.BottonWork}
    source={require('./img/w2.jpg')} />
</TouchableOpacity>
<TouchableOpacity onPress={() => this.goPage('page17')}>
  <Image resizeMode='contain' style={styles.BottonWork}
    source={require('./img/w3.jpg')} />
</TouchableOpacity>
</ScrollView>
const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'column',
    backgroundColor: "#FFCC66",
    padding: 10,
  },
  SmartIMG: {
    width: "70%", height: 80,
    justifyContent: "flex-start",
    marginRight: 5,
    borderColor: '#FF6600',
    borderWidth: 3,
    backgroundColor: '#FFFFFF',
  },
  Logoutbotton: {
    width: "29%", height: 80,

```

```

justifyContent: 'center',
alignItems: 'center',
backgroundColor: '#000000',
borderColor: '#FF6600',
borderWidth: 3
},
BottonWork: {
  backgroundColor: '#FFFFFF',
  width: "100%",
  height: 145,
  marginTop: 10,
  borderColor: '#FFFF99', borderWidth: 2
}
})

```

page3_1.js

```

import React, { Component } from 'react';
import { Select, Option } from "react-native-chooser";
import PickerSelect from 'react-native-picker-select';
import {Platform, StyleSheet, Text, View,
  PanResponder, Image, TouchableOpacity, Alert, ScrollView
} from 'react-native';
import DatePicker from './datepicker.js';
import { callApi } from './api';
export default class page3_1 extends Component {
  constructor(props) {
    super(props);
    this.user_name = "";
    this.state = {
      date: "",
      time: "",
      place: "",
      places: [],
    };
  }
}

```

```

componentDidMount() {
  this.locationList();
}
locationList() {
  let view_url = callApi('location_view.php');
  return fetch(view_url)
    .then((response) => response.json())
    .then((responseJson) => {
      let places = [];
      responseJson.map(function (item, key) {
        places[key] = {}
        places[key].label = item.place_1;
        places[key].value = places[key].label;
      });
      this.setState({ places })
    })
    .catch((error) => {
      console.error(error);
    });
}
onSelect = (place) => {
  //alert(place);
  this.setState({ place });
}
Insertwork = () => {
  const { date } = this.state;
  const { time } = this.state;
  const { place } = this.state;
  let insert_url = callApi('work_insert.php');
  fetch(insert_url, {
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({

```

```

        username: callApi.USERNAME,
        date_1: date,
        time_1: time,
        place_1: place,
    })
  }).then((response) => response.json())
    .then((responseJson) => {
      if (responseJson.status) {
        this.setState({ date: "", time: "", place: "" });
      }
      Alert.alert(responseJson.message);
    }).catch((error) => {
      console.error(error);
    });
}
render() {
  return (
    <ScrollView>
      <View style={styles.container}>
        <View style={{ backgroundColor: '#000033', width: '100%', height: 70, flexDirection:
'row' }}>
          <Text style={{ fontSize: 20, paddingBottom: 20, paddingLeft: 10, paddingTop: 20,
color: 'white' }}>Add Schedule </Text>
        </View>
        <View style={{ paddingTop: 100 }}>
          <DatePicker
            style={{ width: 200 }}
            date={this.state.date}
            mode="date"
            placeholder="เลือกวันที่"
            format="DD-MM-YYYY"
            confirmBtnText="Confirm"
            cancelBtnText="Cancel"
            iconSource={require('./google_calendar.png')}
            onChange={date => this.setState({ date })}

```

```

/>
<Text style={styles.instructions}>date: {this.state.date}</Text>
<DatePicker
  style={{ width: 200 }}
  date={this.state.time}
  mode="time"
  placeholder="เลือกเวลา"
  format="HH:mm"
  confirmBtnText="Confirm"
  cancelBtnText="Cancel"
  minuteInterval={10}
  onChange={time => this.setState({ time })}
/>
<Text style={styles.instructions}>time: {this.state.time}</Text>
<View style={{ flexDirection: 'row', alignItems: 'center'}}>
  <Image source={require('./place.png')} style={{ width: 30, height: 30,
marginLeft: 7, marginRight: 5 }} />
  <View style={{Platform === 'ios' ? null : styles.PickerAndroid}}>
    <PickerSelect
      placeholder={{
        label: 'เลือกสถานที่',
        value: null,
      }}
      hideIcon={true}
      style={pickerSelectStyles}
      value={this.state.place}
      items={this.state.places}
      onChange={(itemValue) => this.setState({ place: itemValue })}
    />
  </View>
</View>
<Text style={styles.instructions}>place: {this.state.place}</Text>
</View>
<TouchableOpacity onPress={this.Insertwork} style={{ width: 120, padding: 10,
backgroundColor: 'black', marginTop: 5, alignItems: 'center', marginLeft: 20 }}>
  <Text style={{ color: 'white' }}>submit</Text>

```

```

        </TouchableOpacity>
        <View style={{ backgroundColor: '#F5FCFF', width: '100%',height:100}}/>
    </View>
</ScrollView>
const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    backgroundColor: '#F5FCFF'
  },
  welcome: {
    fontSize: 20,
    textAlign: 'center',
    margin: 10
  },
  instructions: {
    textAlign: 'center',
    color: '#333333',
    marginBottom: 5
  },
  PickerAndroid: {
    paddingTop: 0,
    paddingLeft: 20,
    paddingBottom: 2,
    borderWidth: 1,
    borderColor: '#cccccc',
    width: 160,
    height:42,
    alignSelf: 'center'
  },
});
const pickerSelectStyles = StyleSheet.create({
  underline:{borderTopWidth:0},
  inputAndroid:{
    transform:[
      {scaleX:0.8},

```

```

        {scaleY:0.8}
    ],
},
inputIOS: {
    paddingTop: 5,
    paddingLeft: 10,
    paddingBottom: 5,
    borderWidth: 0,
    borderColor: '#cccccc',
    width: 160,
    height: 30,
    alignSelf: 'center',
    fontSize: 14,
    color: 'black',
    alignItems: 'center',
}
});

```

page3.js

```

import React, { Component } from 'react';
import {Platform, StyleSheet, Text, View, Image, TextInput, TouchableOpacity, ListView,
ActivityIndicator, Alert, FlatList, AsyncStorage
} from 'react-native';
import { callApi,getLastMonth } from './api';
import Menu from './Menu';
import Dialog from "react-native-dialog";
import { StackActions } from 'react-navigation';
import PickerSelect from 'react-native-picker-select';
export default class page3 extends Component {
    constructor(props) {
        super(props);
        this.state = {isLoading: true, dialogVisible: false,userpassword: "",dialogExport: false,email:
"",month: ""}
        // this.deleteConfirm = this.deleteConfirm.bind(this);
        // this.deleteSubmit = this.deleteSubmit.bind(this);

```

```

// this.selectRow = this.selectRow.bind(this);
this.work_id = 0;
this.action = "";
this.months = getLastMonth();
}
componentDidMount() {
  this.renderList();
}
deleteConfirm(work_id) {
  this.work_id = work_id;
  this.action = 'delete';
  this.setState({ dialogVisible: true });
}
addConfirm() {
  this.action = 'add';
  this.setState({ dialogVisible: true });
}
handleCancel() {
  this.setState({ dialogVisible: false });
};
handleOk() {
  let check_url = callApi('check_password.php');
  fetch(check_url, {
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      username: callApi.USERNAME,
      password: this.state.userpassword,
    })
  }).then((response) => response.json())
  .then((responseJson) => {
    if (responseJson.status) {

```

```

    if(this.action==='delete') {
        this.setState({ dialogVisible: false });
        this.deleteSubmit(this.work_id);
    } else if(this.action==='add') {
        this.setState({ dialogVisible: false });
        this.props.navigation.push('page3_1');
    }
    this.setState({ userpassword: " " });
} else {
    Alert.alert(responseJson.message);
}
}).catch((error) => {
    console.error(error);
});
};
deleteSubmit(work_id) {
    let delete_url = callApi('work_delete.php');
    fetch(delete_url, {
        method: 'POST',
        headers: {
            'Accept': 'application/json',
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({
            username: callApi.USERNAME,
            work_id: work_id,
        })
    }).then((response) => response.json())
    .then((responseJson) => {
        setTimeout(function(){
            Alert.alert(responseJson.message);
        },500);
        this.renderList();
    }).catch((error) => {
        console.error(error);
    });
};

```

```

}
selectRow(work_id,title,latitude,longitude) {
  const pushAction = StackActions.push({
    routeName: 'page4',
    params: {work_id,title,latitude,longitude},
  });

  this.props.navigation.dispatch(pushAction);
}
exportDialog() {
  this.setState({ dialogExport: true });
}
onCloseForm() {
  this.setState({ dialogExport: false });
};
onCancelForm() {
  this.setState({ email: "",month: " " });
};
onSendEmail() {
  this.setState({ dialogLoad: true });
  let check_url = callApi('send_email.php');
  fetch(check_url, {
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      email: this.state.email,
      month: this.state.month,
      email_type: 'PM',
    })
  }).then((response) => response.json())
  .then((responseJson) => {
    if(responseJson.status) {

```

```

        this.onCancelForm();
        this.setState({ dialogExport: false });
    }
    Alert.alert(responseJson.message);
    this.setState({ dialogLoad: false });
}).catch((error) => {
    console.error(error);
    this.setState({ dialogLoad: false });
});
}
renderDialogButton({
    if(this.state.dialogLoad) {
        return (
            <View style={{ flexDirection: 'row', alignItems: 'center', justifyContent:
'center',paddingBottom: 20 }}>
                <Text style={{ fontSize: 16, marginTop: 10,color: '#999999' }}>กำลังดำเนินการ...</Text>
            </View>
        );
    } else {
        return (
            <View style={{ flexDirection: 'row', alignItems: 'center', justifyContent:
'center',paddingBottom: 20 }}>
                <TouchableOpacity onPress={() => this.onSendEmail()} style={{ width: 100, padding:
10, marginRight: 10, backgroundColor: '#3399FF', marginTop: 10, alignItems: 'center', borderRadius:
10 }}>
                    <Text style={{ color: 'white' }}>ส่งอีเมล</Text>
                </TouchableOpacity>
                <TouchableOpacity onPress={() => this.onCloseForm()} style={{ width: 100, padding:
10, backgroundColor: '#CCCCCC', marginTop: 10, alignItems: 'center', borderRadius: 10 }}>
                    <Text style={{ color: 'white' }}>ยกเลิก</Text>
                </TouchableOpacity>
            </View>
        );
    }
}
renderDialog() {

```

```

if (this.state.dialogExport) {
  return (
    <View style={styles.dialogStyle}>
      <View style={styles.dialogForm}>
        <View style={{
          backgroundColor: '#3399FF', width: '100%', height: 35, flexDirection: 'row',
          justifyContent: 'space-between', paddingRight: 30
        }}>
          <Text style={{ fontSize: 19, paddingBottom: 5, paddingLeft: 10, paddingTop: 5,
color: 'white' }}>ส่งอีเมลรายการย้อนหลัง</Text>
        </View>
        <TextInput
          placeholder=""
          onChangeText={email => this.setState({ email })}
          value={this.state.email}
          underlineColorAndroid='transparent'
          style={{ borderWidth: 1, width: 300, margin: 5, height: 40, alignSelf: 'center',
fontSize: 20,paddingLeft: 10 }}
        />
        <View style={{
          backgroundColor: '#3399FF', width: '100%', height: 35, flexDirection: 'row',
          justifyContent: 'space-between', paddingRight: 30
        }}>
          <Text style={{ fontSize: 19, paddingBottom: 5, paddingLeft: 10, paddingTop: 5,
color: 'white' }}
            >รายการย้อนหลัง</Text>
        </View>
        <View style={{Platform==='ios'? null : styles.PickerAndroid}>
          <PickerSelect
            placeholder={{
              label: 'Select Month',
              value: null,
            }}
            hideIcon={true}
            style={pickerSelectStyles}
            value={this.state.month}

```



```

        <TouchableOpacity onPress={() => this.deleteConfirm(rowData.work_id)} style={{
position: 'absolute', backgroundColor: '#6699FF', justifyContent: 'center', alignItems: 'center',
padding: 10, top: 5, bottom: 5, right: 10 }}>
            <Text style={{ color: '#fff', fontSize: 20, }}></Text>
        </TouchableOpacity>
        <Text style={{ paddingLeft: 20, }}>{rowData.place_1}</Text>
    </TouchableOpacity >
)
}
renderList() {
    let view_url = callApi('work_view.php');
    return fetch(view_url)
        .then((response) => response.json())
        .then((responseJson) => {
            let ds = new ListView.DataSource({ rowHasChanged: (r1, r2) => r1 !== r2 });
            this.setState({
                isLoading: false,
                dataSource: ds.cloneWithRows(responseJson),
            }, function () {
                // In this block you can do something with new state.
            });
        })
        .catch((error) => {
            console.error(error);
        });
}
ListViewItemSeparator = () => {
    return (
        <View style={{ width: "100%", margin: 1, }} />
    );
}
render() {
    const { navigation } = this.props;
    if (this.state.isLoading) {
        return (
            <View style={{ flex: 1, paddingTop: 20 }}>

```

```

        <ActivityIndicator />
    </View>
);
}
let inputBorder = 0;
if(Platform.OS !== 'ios' ) {
    inputBorder = 1;
}
return (
    <View style={{ flex: 1, backgroundColor: '#FFcc66' }}>
        <View style={{ backgroundColor: '#9966FF', width: '100%', height: 70, flexDirection: 'row'
    }}>
            <Text style={{ width:'55%',fontSize: 20, paddingBottom: 20, paddingLeft: 10,
paddingTop: 20, color: 'white' }}>PM </Text>
            <View style={{ width:'45%',marginTop: 0,flexDirection: 'row'}}>
                <TouchableOpacity onPress={() => this.addConfirm()} style={{ backgroundColor:
'#000033', width: 35, height: 35, borderRadius: 35, justifyContent: 'center', alignItems: 'center',
elevation: 10, marginLeft: 5, marginTop: 20 }}>
                    <Text style={{ color: '#fff', fontSize: 20 }}>+</Text>
                </TouchableOpacity>
                <TouchableOpacity onPress={() => this.deleteConfirm(0)} style={{
backgroundColor: '#6699FF', width: 35, height: 35, borderRadius: 35, justifyContent: 'center',
alignItems: 'center', elevation: 10, marginLeft: 5, marginTop: 20 }}>
                    <Text style={{ color: '#fff', fontSize: 20 }}>-</Text>
                </TouchableOpacity>
                <TouchableOpacity onPress={() => { this.exportDialog() }} style={{ borderRadius:
20, height: 40,padding: 10, backgroundColor: '#CC3333', alignItems: 'center', justifyContent:
'center',marginLeft: 5, marginTop: 15 }}>
                    <Text style={{ color: 'white' }}>ส่งออก</Text>
                </TouchableOpacity>
            </View>
        </View>
    </View>
    <View style={styles.MainContainer}>
        <ListView
            dataSource={this.state.dataSource}
            renderSeparator={this.ListViewItemSeparator}

```

```

        enableEmptySections={true}
        renderRow={this.renderRow.bind(this)} />
    </View>
    <Menu navigation={this.props.navigation} />
    <Dialog.Container visible={this.state.dialogVisible}>
        <Dialog.Title>Warning</Dialog.Title>
        <Dialog.Description>
            Confirm your password.
        </Dialog.Description>
        <Dialog.Input secureTextEntry
            value={this.state.userpassword}
            underlineColorAndroid='transparent'
            style={{borderBottomWidth:inputBorder}}
            onChangeText={userpassword => this.setState({ userpassword })}
        />
        <Dialog.Button label="Cancel" onPress={() =>this.handleCancel()} />
        <Dialog.Button label="Ok" onPress={() =>this.handleOk()} />
    </Dialog.Container>
    {this.renderDialog()}
</View>
)
}
}
const styles = StyleSheet.create({
  MainContainer: {
    justifyContent: 'center',
    flex: 1,
    margin: 3,
    marginBottom : 80,
  },
  PickerAndroid: {
    paddingTop: 0,
    paddingLeft: 10,
    paddingBottom: 2,
    borderWidth: 1,
    width: 300,

```

```

margin: 5,
alignSelf: 'center',
color: 'black',
},
dialogStyle : {
  position: 'absolute',
  top: 0,
  left: 0,
  height: '100%',
  width: '100%',
  backgroundColor: 'rgba(0,0,0,0.8)',
},
dialogForm : {
  backgroundColor: '#FFFFFF',
  padding: 10,
  borderRadius: 10,
  margin: 10,
  marginTop: 50,
}
});
const pickerSelectStyles = StyleSheet.create({
  inputIOS: {
    paddingTop: 5,
    paddingLeft: 10,
    paddingBottom: 5,
    borderWidth: 0,
    width: 300,
    margin: 5,
    height: 40,
    alignSelf: 'center',
    fontSize: 20,
    color: 'black',
  },
});

```

page4.js

```
import React, { Component } from 'react';
import { Dimensions, StyleSheet, Platform, View, TouchableOpacity, Image, Text } from 'react-native';
import MapView from 'react-native-maps';
import Polyline from '@mapbox/polyline';
import { StackActions } from 'react-navigation';
import { Popup } from 'react-native-map-link';
const { width, height } = Dimensions.get('window');
const ASPECT_RATIO = width / height;
const LATITUDE_DELTA = 0.1922;
const LONGITUDE_DELTA = LATITUDE_DELTA * ASPECT_RATIO;
let GOOGLE_MAPS_APIKEY = "";
if (Platform.OS === 'ios') {
  GOOGLE_MAPS_APIKEY = 'AlzaSyC3lK9mUvFmVuRZytuMO6M9ZqM457VcpxM';
} else {
  GOOGLE_MAPS_APIKEY = 'AlzaSyBiXuqavKNK3dTjySOkAUrp4BlngumPnoI';
}
//13.868685,100.609614
console.disableYellowBox = true;
export default class page4 extends React.Component {
  constructor(props) {
    super(props);
    const { navigation } = this.props;
    this._map = null;
    this.work = {};
    this.work.work_id = navigation.getParam("work_id");
    this.work.title = navigation.getParam("title");
    this.work.initial = {
      latitude: parseFloat(navigation.getParam("latitude")),
      longitude: parseFloat(navigation.getParam("longitude")),
      latitudeDelta: LATITUDE_DELTA,
      longitudeDelta: LONGITUDE_DELTA
    };
  };
  this.state = {
    coords: [],
```

```

    current: {latitude: 0,longitude: 0},
    updateMap : false,
    initialPosition: this.work.initial,
    isVisible: false,
  };
}
componentDidMount() {
  this.getCurrentPosition();
  this._interval = setInterval(() => {
    this.watchPosition();
  }, 10000);
}
componentWillUnmount() {
  navigator.geolocation.clearWatch(this.watchID)
  clearInterval(this._interval);
}
onMapPress = (e) => {
  }
_onRegionChange = (initialPosition) => {
  this.setState({ initialPosition });
}
getCurrentPosition()
{
  navigator.geolocation.getCurrentPosition(
    (position) => {
      this.getDirections(position);
    },
    (error) => console.log(JSON.stringify(error)),
    { enableHighAccuracy: true, timeout: 20000, maximumAge: 1000 },
  );
}
watchPosition()
{this.watchID = navigator.geolocation.watchPosition(
  (position) => {
    this.getDirections(position);},

```

```

(error) => console.log(JSON.stringify(error)),
{ enableHighAccuracy: true, timeout: 20000, maximumAge: 1000 },
);
}
async getDirections(position) {
  try {
    let latitude = (position.coords.latitude);
    let longitude = (position.coords.longitude);
    let current = { latitude, longitude };
    let work = this.work;
    this.setState({current, updateMap: true});
    let startLoc = latitude + "," + longitude;
    let destinationLoc = work.initial.latitude + "," + work.initial.longitude;
    let resp = await fetch('https://maps.googleapis.com/maps/api/directions/json?origin=${
startLoc }&destination=${ destinationLoc }')
    let respJson = await resp.json();
    let points = Polyline.decode(respJson.routes[0].overview_polyline.points);
    let coords = points.map((point, index) => {
      return {
        latitude : point[0],
        longitude : point[1]
      }
    })
    this.setState({coords: coords})
    return coords
  } catch(error) {
    //console.log(error)
    return error
  }
}
}
onPressZoomIn() {
  let {initialPosition} = this.state;
  region = {
    latitude: initialPosition.latitude,
    longitude: initialPosition.longitude,
    latitudeDelta: initialPosition.latitudeDelta / 3,

```

```

        longitudeDelta: initialPosition.longitudeDelta / 3
    };
    this.setState({initialPosition,region});
    this._map.animateToRegion(region, 100);
}
onPressZoomOut() {
    let {initialPosition} = this.state;
    region = {
        latitude: initialPosition.latitude,
        longitude: initialPosition.longitude,
        latitudeDelta: initialPosition.latitudeDelta * 3,
        longitudeDelta: initialPosition.longitudeDelta * 3
    }
    if(region.latitudeDelta<200) {
        this.setState({initialPosition,region});
        this._map.animateToRegion(region, 100);
    }
}
onPressMoreDetail() {
    const pushAction = StackActions.push({
        routeName: 'page5',
        params: {work_id:this.work.work_id,title:this.work.title},
    });
    this.props.navigation.dispatch(pushAction);
}
markerClick() {
    this.setState({isVisible:true});
}
render() {
    let {current} = this.state;
    let work = this.work;
    let Marker = null;
    if(this.state.updateMap) {
        Marker = (

```

```

    <MapView.Marker key={2} coordinate={{latitude: current.latitude, longitude:
current.longitude}}>
      <View><Image source={require('./img/point.gif')} style={{width: 30, height: 30}}
/></View>
    </MapView.Marker>
  );
}
return (
  <View style={styles.MainContainer}>
    <Image source={require('./img/point.gif')} style={{height:0,width:0}}/>
    <Image source={require('./img/work.png')} style={{height:0,width:0}}/>
    <MapView
      ref={component => {this._map = component;}}
      region={this.state.initialPosition}
      onRegionChangeComplete={this._onRegionChange}
      style={StyleSheet.absoluteFill}
      onPress={this.onMapPress}
    >
      <MapView.Marker style={{width: 30, height: 30}} onPress={()=>this.markerClick()}
key={1} title={work.title} coordinate={{latitude: work.initial.latitude, longitude:
work.initial.longitude}}>
        <View>
          <Image source={require('./img/work.png')} style={{width: 30, height: 30}} />
        </View>
      </MapView.Marker>
      {Marker}
      <MapView.Polyline
        coordinates={this.state.coords}
        strokeWidth={2}
        strokeColor="red"/>
    </MapView>
    <TouchableOpacity onPress={()=>{this.onPressZoomIn()}} style={styles.ZoomInStyle}>
      <Text style={styles.ZoomTextStyle}>+</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={()=>{this.onPressZoomOut()}}
style={styles.ZoomOutStyle}>

```

```

        <Text style={styles.ZoomTextStyle}>-</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={()=>{this.onPressMoreDetail()}}
style={styles.MoreDetailStyle}>
        <Text style={styles.MoreTextStyle}>ข้อมูลเพิ่มเติม</Text>
    </TouchableOpacity>
    <Popup
        isVisible={this.state.isVisible}
        onCancelPressed={() => this.setState({ isVisible: false })}
        onAppPressed={() => this.setState({ isVisible: false })}
        onBackButtonPressed={() => this.setState({ isVisible: false })}
        modalProps={{ animationIn: 'slideInUp'}}
        appsWhiteList={['google-maps']}
        options={{
            latitude: this.work.initial.latitude,
            longitude: this.work.initial.longitude,
            title: this.work.initial.latitude+' '+this.work.initial.longitude,
            dialogTitle: 'Open in Google Maps',
            cancelText: 'Cancel'
        }}
    />
</View>
);
}
}
const styles = StyleSheet.create({
    MainContainer: {
        justifyContent: 'center',
        flex: 1,
        margin: 3 },
    ZoomInStyle: {
        backgroundColor: 'rgba(255,255,255,0.9)',
        width: 60,
        height: 30,
        borderRadius: 20,
        position: 'absolute',

```

```

top: 10,
left: '33%',
justifyContent: 'center',
alignItems: 'center',
},
ZoomOutStyle: {
  backgroundColor: 'rgba(255,255,255,0.9)',
  width: 60,
  height: 30,
  borderRadius: 20,
  position: 'absolute',
  top: 10,
  right: '33%',
  justifyContent: 'center',
  alignItems: 'center',
},
ZoomTextStyle : {
  color: '#000000',
  fontSize: 20,
  marginTop: -5,
},
MoreDetailStyle : {
  backgroundColor: '#e89b00',
  width:150,
  height: 50,
  borderRadius: 20,
  position: 'absolute',
  bottom: 10,
  left: '30%',
  justifyContent: 'center',
  alignItems: 'center',
},
MoreTextStyle : {
  color: '#000000',
  fontSize: 16,
  marginTop: -5,});

```

page5.js

```
import React, { Component } from 'react';
import { Platform, StyleSheet, Text, View, Image, TouchableOpacity, ListView, ActivityIndicator,
Alert
} from 'react-native';
import { StackActions } from 'react-navigation';
import { callApi } from './api';
export default class page5 extends Component {
  constructor(props) {
    super(props);
    const { navigation } = this.props;
    this.work_id = navigation.getParam("work_id");
    this.title = navigation.getParam("title");
    //this.work_id = 2;
    this.state = {isLoading: true}
    this.checkDevice = this.checkDevice.bind(this);
    this.changelImage = this.changelImage.bind(this);
  }
  componentDidMount() {
    this.renderList();
  }
  checkDevice(device_id,device_type) {
    const pushAction = StackActions.push({
      routeName: 'page6',
      params: {device_id:device_id,work_id:this.work_id,device_type:device_type},
    });
    this.props.navigation.dispatch(pushAction);
  }
  changelImage(room_id) {
  }
  renderList() {
    let view_url = callApi('building_view.php');
    return fetch(view_url, {
      method: 'POST',
      headers: {
```

```

    'Accept': 'application/json',
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    work_id: this.work_id,
  })
}).then((response) => response.json())
.then((responseJson) => {
  let ds = new ListView.DataSource({ rowHasChanged: (r1, r2) => r1 !== r2 });
  this.setState({
    isLoading: false,
    dataSource: ds.cloneWithRows(responseJson),
  }, function () {
    // In this block you can do something with new state.
  });
}).catch((error) => {
  console.error(error);
});
}
renderRow(rowData,sid,key) {
  return (
    <View key={key} style={{ backgroundColor: 'white' }} >
      <Text style={{ paddingLeft: 10,paddingRight: 10, fontSize: 17, backgroundColor:
'#FFcc66',alignSelf: 'flex-start' }}>ชั้น {rowData.floor} {rowData.building_name} ห้อง
{rowData.room_name}</Text>
      {this.renderDevice(rowData.devices)}
      <View style={styles.imageStyle}>
        {this.renderImage(rowData.room_pic)}
      </View>
    </View>
  )
}
renderDevice(devices) {
  let obj = this;
  return devices.map(function (item,key) {
    return (

```

```

<View key={key} style={{ flex: 1, flexDirection: 'row',backgroundColor: '#FFEBCD', height:
30, marginLeft: 10,marginRight: 10,marginTop: 5, alignSelf: 'center', }} >
  <Text style={{ width:'30%',paddingTop:5,paddingLeft: 3,paddingRight: 3, fontSize: 12
}}>{item.device_name}</Text>
  <Text style={{ width:'30%',paddingTop:5,paddingLeft: 3,paddingRight: 3, fontSize: 12
}}>{item.device_ip}</Text>
  <Text style={{ width:'30%',paddingTop:5,paddingLeft: 3,paddingRight: 3, fontSize: 12
}}>{item.device_no}</Text>
  <TouchableOpacity onPress={() =>
{obj.checkDevice(item.device_id,item.device_type)}} style={{ width:'10%',paddingLeft:
10,paddingRight: 10 }}>
    <Image
      style={{width:28,height:28}}
      source={require('./img/edit.jpg')}
    />
  </TouchableOpacity>
</View>
); });}
renderImage(room_pic) {
  if(room_pic="" && room_pic=null) {
    let image_url = callApi("images/"+room_pic);
    return (
      <Image
        style={styles.roomStyle}
        source={{uri: image_url}}
      />
    )
  } else {
    return (
      <Image
        style={styles.roomStyle}
        source={require('./img/home.jpg')}
      />)}
}
onCloseWork() {
  Alert.alert(
    'Warning',

```

```

    'Are you sure you want to close work?',
    [
      { text: 'Cancel', onPress: () => null, style: 'cancel' },
      { text: 'OK', onPress: () => this.closeWorkSubmit() },
    ],
    { cancelable: false }
  ) }
closeWorkSubmit() {
  let close_url = callApi('work_close.php');
  fetch(close_url, {
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      username: callApi.USERNAME,
      work_id: this.work_id,
    })
  }).then((response) => response.json())
  .then((responseJson) => {
    Alert.alert(responseJson.message);
  }).catch((error) => {
    console.error(error);
  });
}
ListViewItemSeparator = () => {
  return (
    <View style={{ width: "100%", margin: 1, }} />
  );
}
render() {
  if (this.state.isLoading) {
    return (
      <View style={{ flex: 1, paddingTop: 20 }}>

```

```

        <ActivityIndicator />
    </View>
    );
}
return (
    <View style={{ flex: 1, backgroundColor: '#FFcc66' }}>
        <View style={{ backgroundColor: '#000033', width: '100%', height: 70, flexDirection: 'row'
    }}>
            <Text style={{ width:'80%',fontSize: 20, paddingBottom: 20, paddingLeft: 10,
paddingTop: 20, color: 'white' }}>PM:{this.title} </Text>
            <View style={{ width:'20%',marginTop: 15}}>
                <TouchableOpacity onPress={() => { this.onCloseWork() }} style={{ borderRadius:
20, height: 40,padding: 10, backgroundColor: '#CC3333', alignItems: 'center', justifyContent: 'center',
    }}>
                    <Text style={{ color: 'white' }}>เสร็จงาน</Text>
                </TouchableOpacity>
            </View>
        </View>
        <View style={styles.MainContainer}>
            <ListView
                dataSource={this.state.dataSource}
                renderSeparator={this.ListViewItemSeparator}
                enableEmptySections={true}
                renderRow={this.renderRow.bind(this)} />
        </View>
    </View>
) }}
const styles = StyleSheet.create({
    MainContainer: {
        justifyContent: 'center',
        flex: 1,
        margin: 3
    },
    imageStyle: {
        alignItems: 'flex-end',
        marginTop: 5,

```

```
marginBottom: 5,  
marginRight: 10,  
},  
roomStyle: {  
width: 100,  
height : 100  
}  
});
```





เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

building_cm_view.php

```
<?php
include'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json,true);
$json = array();
if(isset($obj['place_1'])) {
    $place_1 = $obj['place_1'];
    $result = mysqli_query($link,"SELECT
building.building_id,building.building_name,room.room_id,room.floor,room.room_name,room.roo
m_pic
FROM building
JOIN room ON building.building_id = room.building_id
WHERE building.building_status <> '0'
AND room.room_status <> '0'
AND building.place_1 = '$place_1'
GROUP BY
building.building_id,building.building_name,room.room_id,room.floor,room.room_name,room.roo
m_pic
ORDER BY building.building_id,room.floor,room_name ASC");
if(mysqli_num_rows($result)){
while ($row=mysqli_fetch_assoc($result)) {
    $building_id = $row['building_id'];
    $room_id = $row['room_id'];
    $result2 = mysqli_query($link,"SELECT device.*
FROM device
WHERE device.device_status <> '0'
AND device.building_id = '$building_id'
AND device.room_id = '$room_id'
ORDER BY device.device_name ASC");
    $devices = array();
    if(mysqli_num_rows($result2)){
        while ($row2=mysqli_fetch_assoc($result2)) {
            array_push($devices,$row2);
        }
    }
    if(count($devices)>0) {
        $row['devices'] = $devices;
    }
}
```

```

        array_push($json,$row); }}}
mysqli_close($link);
echo json_encode($json);

```

building_view.php

```

<?php
include'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json,true);
$work_id = $obj['work_id'];
$result = mysqli_query($link,"SELECT place_1 FROM work WHERE work_id = '$work_id'");
$rec = mysqli_fetch_assoc($result);
$json = array();
if(isset($rec['place_1'])) {
    $place_1 = $rec['place_1'];
    $result = mysqli_query($link,"SELECT
building.building_id,building.building_name,room.room_id,room.floor,room.room_name,room.roo
m_pic
FROM building
JOIN room ON building.building_id = room.building_id
WHERE building.place_1 = '$place_1'
GROUP BY
building.building_id,building.building_name,room.room_id,room.floor,room.room_name,room.roo
m_pic
ORDER BY building.building_id,room.floor,room_name ASC");
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_assoc($result)) {
        $building_id = $row['building_id'];
        $room_id = $row['room_id'];
        $result2 = mysqli_query($link,"SELECT device.*
FROM device
WHERE device.device_status <> '0'
AND device.building_id = '$building_id'
AND device.room_id = '$room_id'
ORDER BY device.device_name ASC");

```

```

$devices = array();
if(mysqli_num_rows($result2)){
    while ($row2=mysqli_fetch_assoc($result2)) {
        array_push($devices,$row2); }}
if(count($devices)>0) {
    $row['devices'] = $devices;
    array_push($json,$row); }}}
mysqli_close($link);
echo json_encode($json);

```

check_form.php

```

<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$work_id = $obj['work_id'];
$device_id = $obj['device_id'];
$username = $obj['username'];
$createdate = date("Y-m-d H:i:s");
$res = array();
if ($work_id == "" || $device_id == "") {
    $res['message'] = ('Required field');
    $res['status'] = false;
} else {
    $result = mysqli_query($link, "SELECT form_work_id FROM form_work WHERE device_id = '$device_id' AND work_id = '$work_id'");
    if ($result) {
        $row = mysqli_fetch_assoc($result);
        if (isset($row['form_work_id'])) {
            $form_work_id = $row['form_work_id'];
            mysqli_query($link, "UPDATE form_work SET
updateby='$username',lastupdate='$createdate' WHERE form_work_id='$form_work_id'");
        } else {
            $result2 = mysqli_query($link, "SELECT * FROM device WHERE device_id = '$device_id'");
            $row2 = mysqli_fetch_assoc($result2);
            $device_name = $row2['device_name'];

```

```

$device_no = $row2['device_no'];
$device_ip = $row2['device_ip'];
$device_type = $row2['device_type'];
$device_model = $row2['device_model'];
$device_barcode = $row2['device_barcode'];
mysqli_query($link, "INSERT INTO form_work
(work_id,device_id,device_name,device_no,device_ip,device_type,device_model,device_barcode,c
reateby,createdate,updateby,lastupdate)
VALUES('$work_id','$device_id','$device_name','$device_no','$device_ip','$device_type','$device_mo
del','$device_barcode','$username','$createdate','$username','$createdate')");
$form_work_id = mysqli_insert_id($link);
}
$sql_update = "
Label = " . $obj['DeviceStatus1'] . ",
LED_System = " . $obj['DeviceStatus2'] . ",
LED_Power_Supply = " . $obj['DeviceStatus3'] . ",
LED_Fan = " . $obj['DeviceStatus4'] . ",
LED_Line_Card = " . $obj['DeviceStatus5'] . ",
LED_Card_Sup = " . $obj['DeviceStatus6'] . ",
Cable_Connection = " . $obj['DeviceStatus7'] . ",
Air_Conditioner = " . $obj['DeviceStatus8'] . ",
Clean_External_Device = " . $obj['DeviceStatus9'] . ",
Temperature = " . $obj['DeviceStatus10'] . ",
Temperature_txt = " . $obj['DeviceInput10'] . ",
Moisture = " . $obj['DeviceStatus11'] . ",
Moisture_txt = " . $obj['DeviceInput11'] . """;
mysqli_query($link, "UPDATE form_work SET $sql_update WHERE
form_work_id='$form_work_id'");
$res['message'] = ('Save successfully');
$res['status'] = true;
} else {
$res['message'] = ('Save fail!');
$res['status'] = false;
mysqli_close($link);
echo json_encode($res);

```

check_password.php

```
<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$username = $obj['username'];
$password = $obj['password'];
$res = array();
if ($obj['username'] != "" or $obj['password'] != "") {
    $result = mysqli_query($link, "SELECT * FROM login
        WHERE user='$username'
        AND password='$password' and admin='1'");
    if ($result->num_rows == 0) {
        $res['message'] = ('Administrators or Password is either incorrect');
        $res['status'] = false;
    } else {
        $res['message'] = ('ok');
        $res['status'] = true;
    }
} else {
    $res['message'] = ('try again');
    $res['status'] = false;
}
mysqli_close($link);
echo json_encode($res);
```

connections.php

```
<?php
$link = mysqli_connect('127.0.0.1','root','project','project');
mysqli_set_charset($link, "utf8");
if(!$link){
    echo'connections field';
}
```

device_change.php

```
<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$sold_device_name = $obj['old_device_name'];
$sold_device_no = $obj['old_device_no'];
$sold_device_model = $obj['old_device_model'];
$sold_device_barcode = $obj['old_device_barcode'];
$new_device_model = $obj['new_device_model'];
$new_device_no = $obj['new_device_no'];
$new_device_barcode = $obj['new_device_barcode'];
$username = $obj['username'];
$createdate = date("Y-m-d H:i:s");
$result1 = mysqli_query($link, "SELECT store_id,device_no FROM store WHERE device_barcode =
'$new_device_barcode' AND store_status = '0'");
$row1 = mysqli_fetch_assoc($result1);
$result2 = mysqli_query($link, "SELECT device_id,device_no FROM device WHERE device_barcode
= '$sold_device_barcode'");
$row2 = mysqli_fetch_assoc($result2);
$new_store_id = @$row1['store_id'];
$sold_device_id = @$row2['device_id'];
$res = array();
if (!isset($row1['device_no'])) {
    $res['status'] = false;
    $res['message'] = 'No new device found.';
} else if (!isset($row2['device_no'])) {
    $res['status'] = false;
    $res['message'] = 'No old device found.';
} else {
    if ($sold_device_no == "" || $new_device_no == "" || $sold_device_barcode == "" ||
    $new_device_barcode == "") {
        $res['status'] = false;
        $res['message'] = 'Required field';
    } else if (mysqli_query($link, "INSERT INTO device_change
(old_device_id,old_device_name,old_device_model,old_device_no,old_device_barcode,new_stor
```

```

e_id,new_device_model,new_device_no,new_device_barcode,createby,createdate)
VALUES('$old_device_id','$old_device_name','$old_device_model','$old_device_no','$old_device_b
arcode','$new_store_id','$new_device_model','$new_device_no','$new_device_barcode','$usernam
e','$createdate')) {
    mysqli_query($link, "UPDATE device SET
    device_no='$new_device_no',
    device_model='$new_device_model',
    device_barcode='$new_device_barcode',
    updateby='$username',
    lastupdate='$createdate'
    WHERE device_id='$old_device_id'");
    $res['status'] = true;
    $res['message'] = 'Input successsfully';
} else {
    $res['status'] = false;
    $res['message'] = 'Input fail!';
mysqli_close($link);
echo json_encode($res);

```

device_type_view.php

```

<?php
include'connections.php';
$json = array();
$result = mysqli_query($link,"SELECT * FROM device_type ORDER BY device_type_id ASC");
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_assoc($result)) {
        array_push($json,$row);
    }
}
mysqli_close($link);
echo json_encode($json);

```

form_work_view.php

```
<?php
include'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json,true);
$device_id = $obj['device_id'];
$work_id = $obj['work_id'];
$device_type = array("Core switch","Distribution switch","Aggregate switch","Access switch","Router");
$json = array();
$result = mysqli_query($link,"SELECT * FROM form_work WHERE device_id = '$device_id' AND
work_id = '$work_id'");
$row = mysqli_fetch_assoc($result);
if(!isset($row['device_id'])) {
    $result = mysqli_query($link,"SELECT * FROM device WHERE device_id = '$device_id'");
    $row = mysqli_fetch_assoc($result);
} else {
    $row['DeviceStatus1'] = $row['Label'];
    $row['DeviceStatus2'] = $row['LED_System'];
    $row['DeviceStatus3'] = $row['LED_Power_Supply'];
    $row['DeviceStatus4'] = $row['LED_Fan'];
    $row['DeviceStatus5'] = $row['LED_Line_Card'];
    $row['DeviceStatus6'] = $row['LED_Card_Sup'];
    $row['DeviceStatus7'] = $row['Cable_Connection'];
    $row['DeviceStatus8'] = $row['Air_Conditioner'];
    $row['DeviceStatus9'] = $row['Clean_External_Device'];
    $row['DeviceStatus10'] = $row['Temperature'];
    $row['DeviceInput10'] = $row['Temperature_txt'];
    $row['DeviceStatus11'] = $row['Moisture'];
    $row['DeviceInput11'] = $row['Moisture_txt']; }
if(in_array($row['device_type'],$device_type)) {
    $row['type_no'] = intval(array_search($row['device_type'], $device_type));
} else { $row['type_no'] = intval(count($device_type)); }
$json = $row;
mysqli_close($link);
echo json_encode($json);
```

input_device.php

```
<?php
include'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json,true);
$device_barcode = $obj['device_barcode'];
$json = array();
$result = mysqli_query($link, "SELECT * FROM store WHERE device_barcode = '$device_barcode'
AND store_status = '1'");
$row = mysqli_fetch_assoc($result);
if(isset($row['device_no'])) {
    $res['status'] = true;
    $res['message'] = $row;
} else {
    $res['status'] = false;
    $res['message'] = 'No device store found!';
}
mysqli_close($link);
echo json_encode($res);
```

location_view.php

```
<?php
include'connections.php';
$json = array();
$result = mysqli_query($link,"SELECT * FROM location ORDER BY place_1 ASC");
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_assoc($result)) {
        array_push($json,$row);
    }
}
mysqli_close($link);
echo json_encode($json);
```

login.php

```
<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$user = $obj['user'];
$password = $obj['password'];
$res = array();
if ($obj['user'] != "" or $obj['password'] != "") {
    $result = mysqli_query($link, "SELECT * FROM login where user='$user' and
password='$password'");
    if ($result->num_rows == 0) {
        $res['message'] = ('Username or Password is either incorrect!');
        $res['status'] = false;
    } else {
        $res['message'] = ('ok');
        $res['status'] = true; }
} else {
    $res['message'] = ('try again');
    $res['status'] = false; }
echo json_encode($res);?>
```

new_device.php

```
<?php
include'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json,true);
$device_barcode = $obj['device_barcode'];
$json = array();
$result = mysqli_query($link, "SELECT * FROM store WHERE device_barcode = '$device_barcode' ");
$row = mysqli_fetch_assoc($result);
if(isset($row['device_no'])) {
    $res['status'] = true;
    $res['message'] = $row;
} else {
```

```

$res['status'] = false;
$res['message'] = 'No new device found.';
}
mysqli_close($link);
echo json_encode($res);

```

old_device.php

```

<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json,true);
$device_barcode = $obj['device_barcode'];
$res = array();
$result = mysqli_query($link, "SELECT * FROM device WHERE device_barcode =
'$device_barcode'");
$row = mysqli_fetch_assoc($result);
if(isset($row['device_no'])) {
    $res['status'] = true;
    $res['message'] = $row;
} else {
    $res['status'] = false;
    $res['message'] = 'No old device found.';
}
mysqli_close($link);
echo json_encode($res);

```

problem_insert.php

```

<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$device_id = $obj['device_id'];
$problem_date = strtotime(str_replace("/", "-", $obj['problem_date']));
$problem_date = date("Y-m-d", $problem_date);
$problem_no = $obj['problem_no'];

```

```

$problem_type = $obj['problem_type'];
$problem_detail = $obj['problem_detail'];
$problem_edit = $obj['problem_edit'];
$username = $obj['username'];
$createdate = date("Y-m-d H:i:s");
$result = mysqli_query($link, "SELECT * FROM device WHERE device_id = '$device_id'");
$row = mysqli_fetch_assoc($result);
$res = array();
if (isset($row['device_no'])) {
    $res['status'] = false;
    $res['message'] = 'No device found.';
} else {
    $device_name = $row['device_name'];
    $device_no = $row['device_no'];
    $device_ip = $row['device_ip'];
    $device_type = $row['device_type'];
    $device_model = $row['device_model'];
    $device_barcode = $row['device_barcode'];
    mysqli_query($link, "INSERT INTO problem
(device_id,device_name,device_no,device_ip,device_type,device_model,device_barcode,createby,
createdate)
VALUES('$device_id','$device_name','$device_no','$device_ip','$device_type','$device_model','$device_barcode','$username','$createdate')");
    $problem_id = mysqli_insert_id($link);
    $sql_update = "
        problem_date = " . @$problem_date . ",
        problem_no = " . $obj['problem_no'] . ",
        problem_type = " . $obj['problem_type'] . ",
        problem_detail = " . $obj['problem_detail'] . ",
        problem_edit = " . $obj['problem_edit'] . """;
    mysqli_query($link, "UPDATE problem SET $sql_update WHERE problem_id='$problem_id'");
    $res['status'] = true;
    $res['message'] = 'Input successfully';
}
mysqli_close($link);
echo json_encode($res);

```

problem_view.php

```
<?php
include'connections.php';
$json = array();
$result = mysqli_query($link,"SELECT * FROM problem ORDER BY problem_date DESC,problem_id
DESC");
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_assoc($result)) {
        $row['problem_date'] = date("d/m/Y",strtotime($row['problem_date']));
        array_push($json,$row);
    }
}
mysqli_close($link);
echo json_encode($json);
```

send_email.php

```
<?php
include 'connections.php';
$month_name = array("", "มกราคม", "กุมภาพันธ์", "มีนาคม", "เมษายน", "พฤษภาคม", "มิถุนายน", "กรกฎาคม", "สิงหาคม", "กันยายน", "ตุลาคม", "พฤศจิกายน", "ธันวาคม");
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$email = $obj['email'];
$month = $obj['month'];
$email_type = $obj['email_type'];
$res = array();
if ($email == "" || $month == "" || $email_type == "") {
    $res['message'] = ('Required field');
    $res['status'] = false;
} else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $res['message'] = ('Email fail!');
    $res['status'] = false;
} else if ($email_type == "PM") {
    $filename = $email_type.str_replace(' ', '', $month).'.xlsx';
    list($m, $y) = explode(" ", $month);
    $subject = 'รายงาน PM ประจำเดือน '.$month_name[$m].' '.$y;
```

```

$sheets = array();
$datalist = array();
$result = mysqli_query($link,"SELECT * FROM work join location ON work.place_1 =
location.place_1 WHERE DATE_FORMAT(date_1,'%c %Y') = '$month' ORDER BY date_1,time_1
ASC");
if(mysqli_num_rows($result)) {
    array_push($datalist,array('วันที่','เวลา','สถานที่','สถานะ'));
    while ($row=mysqli_fetch_assoc($result)) {
        $row['date_1'] = date("d/m/Y",strtotime($row['date_1']));
        $row['time_1'] = date("Hi",strtotime($row['time_1']));
        $row['work_status'] = ($row['work_status']!= '1') ? 'กำลังดำเนินการ' : 'ปิดงาน';
        $data = array (
            $row['date_1'],
            $row['time_1'],
            $row['place_1'],
            $row['work_status'],
        );
        array_push($datalist,$data); }
    }
array_push($sheets,array("name"=>"PM", "datalist"=>$datalist));
$datalist = array();
$result = mysqli_query($link,"SELECT * FROM form_work WHERE DATE_FORMAT(lastupdate,'%c
%Y')= '$month' ORDER BY lastupdate DESC ");
if(mysqli_num_rows($result)) {
    array_push($datalist,array('วันที่','ชื่ออุปกรณ์','IP','หมายเลขเครื่อง','ชนิดของอุปกรณ์','ป้ายชื่อ','สถานะ
LED system','สถานะ LED Power Supply','สถานะ LED Fan'
, 'สถานะ LED Line card','สถานะ LED Card Sup','การเชื่อมต่อสาย','การทำงานของเครื่องปรับอากาศ','ทำ
ความสะอาด','อุณหภูมิ','สถานะอุณหภูมิ','ความชื้น','สถานะความชื้น','ผู้ทำ'));
    while ($row=mysqli_fetch_assoc($result)) {
        $row['lastupdate'] = date("d/m/Y",strtotime($row['lastupdate']));
        $data = array (
            $row['lastupdate'],
            $row['device_name'],
            $row['device_ip'],
            $row['device_no'],
            $row['device_type'],

```

```

$row['Label'],
$row['LED_System'],
$row['LED_Power_Supply'],
$row['LED_Fan'],
$row['LED_Line_Card'],
$row['LED_Card_Sup'],
$row['Cable_Connection'],
$row['Air_Conditioner'],
$row['Clean_External_Device'],
$row['Temperature_txt'],
$row['Temperature'],
$row['Moisture_txt'],
$row['Moisture'],
$row['updateby'],
);
array_push($datalist,$data); }}
array_push($sheets,array("name"=>"สถานะ","datalist"=>$datalist)); //coppyy fix
create_excel($filename,$sheets);
$res = send_email($filename,$subject,$email);
} else if ($email_type == "CM") {
    $filename = $email_type.str_replace(",",$month).!xlsx;
    list($m,$y) = explode(" ",$month);
    $subject = 'รายงานปัญหาที่เกิดขึ้นแล้ว ประจำเดือน '.$month_name[$m].' '.$y;
    $sheets = array();
    $datalist = array();
    $result = mysqli_query($link,"SELECT * FROM problem WHERE
DATE_FORMAT(problem_date,'%c %Y') = '$month' ORDER BY problem_date DESC,problem_id
DESC ");
    if(mysqli_num_rows($result)) {
        array_push($datalist,array('วันที่','เลขเคส','ชื่ออุปกรณ์','Serial no','IP','Barcode','Model','Type','
ปัญหา','วิธีแก้'));
        while ($row=mysqli_fetch_assoc($result)) {
            $row['problem_date'] = date("d/m/Y",strtotime($row['problem_date']));
            $data = array (
                $row['problem_date'],
                $row['problem_no'],

```

```

        $row['device_name'],
        $row['device_no'],
        $row['device_ip'],
        $row['device_barcode'],
        $row['device_model'],
        $row['problem_type'],
        $row['problem_detail'],
        $row['problem_edit']
    );
    array_push($datalist,$data); }}
array_push($sheets,array("name"=>"รายงานปัญหาที่เกิดขึ้นแล้ว","datalist"=>$datalist));
create_excel($filename,$sheets);
$res = send_email($filename,$subject,$email);
} else if ($email_type == "STORE") {
    $filename = $email_type.str_replace(' ','',$month).'.xlsx';
    list($m,$y) = explode(" ",$month);
    $subject = 'รายงานการคลังพัสดุ ประจำเดือน '.$month_name[$m].' '.$y;
    $sheets = array();
    $datalist = array();
    $result = mysqli_query($link,"SELECT COUNT(*) AS device_count,store.device_type
    FROM store JOIN device_type dt ON dt.device_type_name = store.device_type
    WHERE DATE_FORMAT(store_date,'%c %Y') = '$month' AND store.store_status='1'
    GROUP BY device_type ORDER BY dt.device_type_id ASC");
    if(mysqli_num_rows($result)) {
        array_push($datalist,array('Type','จำนวน'));
        while ($row=mysqli_fetch_assoc($result)) {
            $data = array (
                $row['device_type'],
                $row['device_count'],);
            array_push($datalist,$data); }
        array_push($sheets,array("name"=>"คลังพัสดุ","datalist"=>$datalist));
        $datalist = array();
        $result = mysqli_query($link,"SELECT * FROM store WHERE DATE_FORMAT(store_date,'%c %Y')
        = '$month' ORDER BY store_date DESC ,createdate DESC ");
        if(mysqli_num_rows($result)) {

```

```

        array_push($datalist,array('วันที่','Serial no','Barcode','Model','Type','สถานที่','สถานะ','ผู้นำเข้า/ออก',
        หมายเหตุ));
        while ($row=mysqli_fetch_assoc($result)) {
            $row['store_date'] = date("d/m/Y",strtotime($row['store_date']));
            $row['store_status'] = ($row['store_status']=='1' || $row['store_status']=='0') ? 'นำเข้า' : 'นำ
        ออก';

            $data = array (
                $row['store_date'],
                $row['device_no'],
                $row['device_barcode'],
                $row['device_model'],
                $row['device_type'],
                $row['place_1'],
                $row['store_status'],
                $row['store_by'],
                $row['store_remark'],);
            array_push($datalist,$data);
        array_push($sheets,array("name"=>"นำเข้า-นำออก","datalist"=>$datalist));
        create_excel($filename,$sheets);
        $res = send_email($filename,$subject,$email);
    } else {
        $res['message'] = ('Email Type fail!');
        $res['status'] = false; }
    mysqli_close($link);
    echo json_encode($res);
    function create_excel($filename = ",$sheets = array()) {
        include 'libraries/PHPExcel.php';
        $objPHPExcel = new PHPExcel();
        foreach($sheets as $i=>$sheet) {
            if($i==0) {
                $objPHPExcel->setActiveSheetIndex(0);
            } else {
                $objPHPExcel->createSheet($i);
                $objPHPExcel->setActiveSheetIndex($i); }
            $objPHPExcel->getActiveSheet()->setTitle($sheet['name']);
            $rows = 1;

```

```

foreach($sheet['datalist'] as $key=>$val) {
    $cols = 'A';
    foreach($val as $key2=>$val2) {
        $objPHPExcel->getActiveSheet()->setCellValue($cols.$rows, $val2);
        $cols++;
    }
    $rows++;
}
$objWriter = PHPExcel_IOFactory::createWriter($objPHPExcel, 'Excel2007');
$objWriter->save("files/$filename");
function send_email($filename = "", $subject = "", $email = "")
{
    $result = array();
    include 'libraries/PHPMailer/smtp.php';
    include 'libraries/PHPMailer/phpmailer.php';
    $mail = new PHPMailer();
    $mail->isSMTP();
    $mail->Host = 'ssl://smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'psmart.app@gmail.com';
    $mail->Password = 'Psmart2018';
    $mail->SMTPSecure = 'ssl';
    $mail->Port = '465';
    $mail->CharSet="utf-8";
    $mail->setFrom('psmart.app@gmail.com');
    $mail->addAddress($email);
    $mail->addAttachment("files/$filename");
    $mail->isHTML(true);
    $mail->Subject = $subject;
    $mail->Body = $subject;
    if(!$mail->send()) {
        $result['status'] = false;
        $result['message'] = $mail->ErrorInfo;
    } else {
        $result['status'] = true;
        $result['message'] = "Send email successfully";
    }
    return $result;
}
}

```

store_device.php

```
<?php
include'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json,true);
$device_type = $obj['device_type'];
$json = array();
$result = mysqli_query($link,"SELECT * FROM store WHERE device_type='$device_type' AND
store_status='1' ORDER BY device_model ASC");
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_assoc($result)) {
        array_push($json,$row);
    }
}
mysqli_close($link);
echo json_encode($json);
```

store_input.php

```
<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$device_no = $obj['device_no'];
$device_model = $obj['device_model'];
$device_type = $obj['device_type'];
$device_barcode = $obj['device_barcode'];
$store_by = $obj['store_by'];
$store_date = strtotime(str_replace("/", "-", $obj['store_date']));
$store_date = date("Y-m-d", $store_date);
$store_remark = $obj['store_remark'];
$username = $obj['username'];
$createdate = date("Y-m-d H:i:s");
$result = mysqli_query($link, "SELECT device_no FROM store WHERE device_barcode =
'$device_barcode' AND store_status='1' ");
$row = mysqli_fetch_assoc($result);
```

```

$res = array();
if (isset($row['device_no'])) {
    $res['message'] = ('Input barcode duplicate. ');
    $res['status'] = false;
} else {
    if ($device_barcode == "" || $device_no == "" || $device_model == "" || $device_type == "" ||
    $store_by == "" || $obj['store_date'] == "") {
        $res['message'] = ('Required field');
        $res['status'] = false;
    } else if (mysqli_query($link, "INSERT INTO store
(device_no,device_model,device_type,device_barcode,store_status,store_remark,store_by,store_d
ate,createby,createdate)
VALUES('$device_no','$device_model','$device_type','$device_barcode','1','$store_remark','$store_by',
'$store_date','$username','$createdate')") {
        $res['message'] = ('Input successfully');
        $res['status'] = true;
    } else {
        $res['message'] = ('Input fail');
        $res['status'] = false; }}
mysqli_close($link);
echo json_encode($res);

```

store_output.php

```

<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$place_1 = $obj['place_1'];
$device_no = $obj['device_no'];
$device_model = $obj['device_model'];
$device_type = $obj['device_type'];
$device_barcode = $obj['device_barcode'];
$store_by = $obj['store_by'];
$store_date = strtotime(str_replace("/", "-", $obj['store_date']));
$store_date = date("Y-m-d", $store_date);
$store_remark = $obj['store_remark'];

```

```

$username = $obj['username'];
$createdate = date("Y-m-d H:i:s");
$result1 = mysqli_query($link, "SELECT store_id,device_no FROM store WHERE device_barcode =
'$device_barcode' AND store_status = '1'");
$row1 = mysqli_fetch_assoc($result1);
// $result2 = mysqli_query($link, "SELECT store_id,device_no FROM store WHERE device_barcode
= '$device_barcode' AND store_status = '0'");
// $row2 = mysqli_fetch_assoc($result2);
$res = array();
if (isset($row1['device_no'])) {
    $res['message'] = ('No device store found. ');
    $res['status'] = false;
// } else if (isset($row2['device_no'])) {
//     $res['message'] = ('Output barcode duplicate. ');
//     $res['status'] = false;
} else {
    if ($device_barcode == "" || $device_no == "" || $device_model == "" || $device_type == "" ||
$store_by == "" || $obj['store_date'] == "") {
        $res['message'] = ('Required field');
        $res['status'] = false;
    } else if (mysqli_query($link, "INSERT INTO store
(place_1,device_no,device_model,device_type,device_barcode,store_status,store_remark,store_by,
store_date,createby,createdate)
VALUES('$place_1','$device_no','$device_model','$device_type','$device_barcode','0','$store_remark'
,$store_by','$store_date','$username','$createdate')")) {
        $store_id = $row1['store_id'];
        mysqli_query($link,"UPDATE store SET store_status = '2' WHERE store_id='$store_id'");
        $res['message'] = ('Input successsfully');
        $res['status'] = true;
    } else {
        $res['message'] = ('Input fail!');
        $res['status'] = false;
    }
}
mysqli_close($link);
echo json_encode($res);

```

store_summary.php

```
<?php
include'connections.php';
$json = array();
$result = mysqli_query($link,"SELECT COUNT(*) AS device_count,store.device_type
FROM store JOIN device_type dt ON dt.device_type_name = store.device_type
WHERE store.store_status='1'
GROUP BY device_type
ORDER BY dt.device_type_id ASC");
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_assoc($result)) {
        array_push($json,$row); }
mysqli_close($link);
echo json_encode($json);
```

store_view.php

```
<?php
include'connections.php';
$json = array();
$result = mysqli_query($link,"SELECT * FROM store ORDER BY store_date DESC ,createdate DESC");
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_assoc($result)) {
        $row['store_date'] = date("d/m/Y",strtotime($row['store_date']));
        $row['store_status'] = ($row['store_status']=='2') ? '1' : $row['store_status'];
        array_push($json,$row);}
mysqli_close($link);
echo json_encode($json);
```

users_view.php

```
<?php
include'connections.php';
$json = array();
$result = mysqli_query($link,"SELECT * FROM users ORDER BY firstname,lastname ASC");
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_assoc($result)) {
```

```
        array_push($json,$row); }}
mysqli_close($link);
echo json_encode($json);
```

work_close.php

```
<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$work_id = $obj['work_id'];
$username = $obj['username'];
$lastupdate = date("Y-m-d H:i:s");
$res = array();
if ($work_id == 0) {
    $res['message'] = ('Close fail');
    $res['status'] = false;
} else {
    if (mysqli_query($link, "UPDATE work SET
work_status='2',updateby='$username',lastupdate='$lastupdate' WHERE work_id ='" . $work_id . "'"))
    {
        $res['message'] = ('Close work successfully');
        $res['status'] = true;
    } else {
        $res['message'] = ('Close fail');
        $res['status'] = false;
    }
}
mysqli_close($link);
echo json_encode($res);
```

work_delete.php

```
<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$work_id = $obj['work_id'];
$username = $obj['username'];
$res = array();
if ($work_id == 0) {
    if (mysqli_query($link, "DELETE FROM work") {
        $res['message'] = ('DELETE successfully');
        $res['status'] = true;
    } else {
        $res['message'] = ('DELETE fail');
        $res['status'] = false;
    }
} else {
    if (mysqli_query($link, "DELETE FROM work WHERE work_id = " . $work_id . " ") {
        $res['message'] = ('DELETE successfully');
        $res['status'] = true;
    } else {
        $res['message'] = ('DELETE fail');
        $res['status'] = false;
    }
}
mysqli_close($link);
echo json_encode($res);
```

work_insert.php

```
<?php
include 'connections.php';
$json = file_get_contents('php://input');
$obj = json_decode($json, true);
$date = strtotime(str_replace("/", "-", $obj['date_1']));
$date = date("Y-m-d", $date);
$time = $obj['time_1'];
$place = $obj['place_1'];
```

```

$username = $obj['username'];
$createdate = date("Y-m-d H:i:s");
$res = array();
if ($date == "" || $time == "" || $place == "") {
    $res['message'] = ('Required field');
    $res['status'] = false;
} else if (mysqli_query($link, "INSERT INTO work (date_1,time_1,place_1,createby,createdate)
VALUES('$date','$time','$place','$username','$createdate')")) {
    $res['message'] = ('Insert successsfully');
    $res['status'] = true;
} else {
    $res['message'] = ('Insert fail!');
    $res['status'] = false;
}
mysqli_close($link);
echo json_encode($res);

```

work_view.php

```

<?php
include'connections.php';
$json = array();
$result = mysqli_query($link,"SELECT * FROM work join location ON work.place_1 =
location.place_1 ORDER BY date_1,time_1 ASC");
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_assoc($result)) {
        $row['date_1'] = date("d/m/Y",strtotime($row['date_1']));
        $row['time_1'] = date("H:i",strtotime($row['time_1']));
        array_push($json,$row);
    }
}
mysqli_close($link);
echo json_encode($json);

```