



การพัฒนาโปรแกรมบริหารจัดการบำรุงรักษาอาคาร



โดย  
นายมาย ยินชัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาคอมพิวเตอร์เพื่อการออกแบบทางสถาปัตยกรรม แผน ก แบบ ก 2 ระดับปริญญา

มหาบัณฑิต

ภาควิชาเทคนิคสถาปัตยกรรม

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2560

ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

# การพัฒนาโปรแกรมบริหารจัดการบำรุงรักษาอาคาร



โดย  
นายมาย ยืนชัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาคอมพิวเตอร์เพื่อการออกแบบทางสถาปัตยกรรม แผนก ก แบบ ก 2 ระดับปริญญา

มหาบัณฑิต

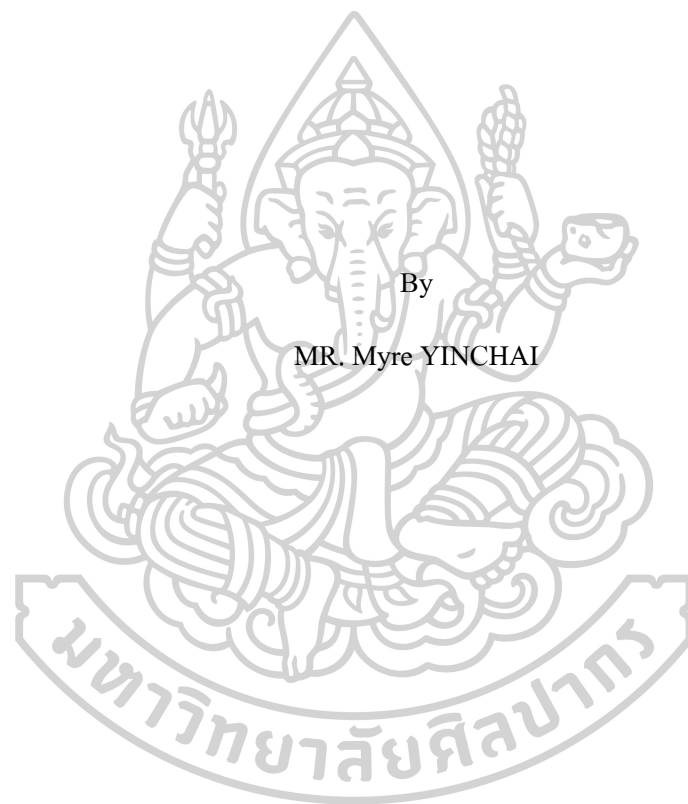
ภาควิชาเทคนิคสถาปัตยกรรม

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2560

ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

FACILITIES OPERATIONS AND MAINTENANCE SOFTWARE  
DEVELOPMENT



By  
MR. Myre YINCHAI

A Thesis Submitted in Partial Fulfillment of the Requirements  
for Master of Science (Computer-aided Architectural Design)

Department of Architectural Technology

Graduate School, Silpakorn University

Academic Year 2017

Copyright of Graduate School, Silpakorn University

หัวข้อ	การพัฒนาโปรแกรมบริหารจัดการบำรุงรักษาอาคาร
โดย	มาย ยินชัย
สาขาวิชา	คอมพิวเตอร์เพื่อการออกแบบทางสถาปัตยกรรม แผนก ก แบบ ก 2 ระดับปริญญาโท
อาจารย์ที่ปรึกษาหลัก	ศาสตราจารย์ จูติพัฒน์ ประทานทรัพย์

---

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร ได้รับพิจารณาอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต

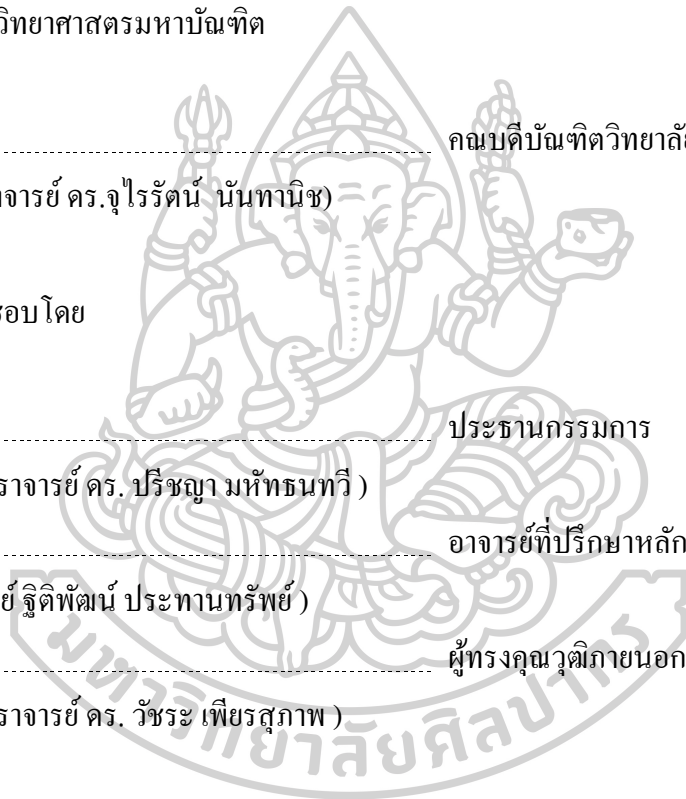
..... คณบดีบัณฑิตวิทยาลัย  
(รองศาสตราจารย์ ดร. จุไรรัตน์ นันทนิช)

พิจารณาเห็นชอบ โดย

..... ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. ปรีชญานันท์ มัทธนทวี)

..... อาจารย์ที่ปรึกษาหลัก  
(ศาสตราจารย์ จูติพัฒน์ ประทานทรัพย์)

..... ผู้ทรงคุณวุฒิภายนอก  
(ผู้ช่วยศาสตราจารย์ ดร. วัชรเกียรติ สุภาพ)



56059310 : คอมพิวเตอร์เพื่อการออกแบบทางสถาปัตยกรรม แผน ก แบบ ก 2 ระดับปริญญา  
มหาบัณฑิต

คำสำคัญ : ซ่อมบำรุงรักษาอาคาร, แอ้งซ่อม

นาย นาย ยินชัย: การพัฒนาโปรแกรมบริหารจัดการบำรุงรักษาอาคาร อาจารย์ที่ปรึกษา  
วิทยานิพนธ์ : ศาสตราจารย์ จูติพัฒน์ ประทานทรัพย์

งานดูแลอาคาร (Building Maintenance) จะเกิดขึ้นภายหลังเมื่ออาคารก่อสร้างแล้ว  
เสร็จเมื่อมีการเข้าใช้อาคาร ผู้บริหารจัดการทรัพยากรอาคารมีหน้าที่ในการตรวจสอบสภาพการใช้  
งานของสาธารณูปโภคต่าง ๆ ในอาคาร ทุกงานเพื่อให้ระบบทำงานได้อย่างมีประสิทธิภาพ ซึ่ง  
โดยทั่วไปอาคารต่างๆ จะมีหน่วยซ่อมบำรุงประจำอาคารเพื่อคอยแก้ไขปัญหาต่างๆ ที่จะ  
เกิดขึ้น การซ่อมบำรุงอาคาร จะใช้กระบวนการบันทึกข้อมูลต่างๆลงกระดาษแบบฟอร์มหรือการ  
ติดต่อผ่านโทรศัพท์เพื่อแจ้งปัญหาที่เกิดขึ้น กระบวนการลักษณะนี้มักจะเกิดปัญหาล่าช้าในการส่ง  
เอกสาร เอกสารสูญหาย มีความยากลำบากในการติดตามความคืบหน้าของงาน และยังขาด  
ประสิทธิภาพในการรวบรวมข้อมูลการซ่อมต่างๆ

เนื่องจากในปัจจุบันเทคโนโลยีได้เข้ามามีบทบาทต่อการทำงานอย่างมากซึ่งก็สามารถ  
ช่วยให้การทำงานต่างๆรวดเร็ว สะดวก ถูกต้อง มากยิ่งขึ้น โดยได้มีการนำเทคโนโลยีมารวมกับ  
แนวความคิดในการบริหารทรัพยากรอาคาร เพื่อช่วยให้การจัดการข้อมูล และการจัดการระบบการ  
ทำงานต่างๆ มีประสิทธิภาพมากขึ้น

จึงเป็นที่มาของการพัฒนาโปรแกรมสำหรับบำรุงรักษาซ่อมแซมอาคาร เพื่อเป็น  
ทางเลือกสำหรับผู้บริหารจัดการทรัพยากรอาคาร โดยโปรแกรมมีการเก็บข้อมูลต่างๆ ที่เหมาะสม  
กับลักษณะงานตามโครงสร้างขององค์กร ซึ่งการจัดการอย่างเป็นระบบผ่าน โปรแกรมคอมพิวเตอร์  
จะช่วยให้ ระบบสาธารณูปโภคภายในอาคารมีอายุการใช้งานที่ยาวนานขึ้นและยังเป็นการป้องกัน  
อุบัติเหตุที่อาจเกิดขึ้นจากความเสื่อมของอุปกรณ์ภายในอาคาร อีกทั้งการนำโปรแกรมบริหาร  
จัดการบำรุงรักษาอาคารมาใช้จะช่วยลดข้อผิดพลาดต่างๆ ให้สามารถแสดงรายงานสถานะของงาน  
และยังสามารถนำข้อมูลไปใช้ในการวิเคราะห์และรวมถึงการวางแผนงบประมาณได้อีกด้วย

56059310 : Major (Computer-aided Architectural Design)

Keyword : Building Maintenance, WorkOrder, CAFM

MR. MYRE YINCHAI : FACILITIES OPERATIONS AND MAINTENANCE  
SOFTWARE DEVELOPMENT THESIS ADVISOR : PROFESSOR THITIPAT  
PRATHARNSAP

Building Maintenance will typically take place after the building construction has been completed, and all facilities are fully accessible. The Building Resource Manager is responsible for inspecting the use of public utilities, and to ensure all systems work efficiently. In general, there will be repair units appointed for maintaining and resolving technical issues. The process of building maintenance relies on manual problem logging and trouble ticketing procedures, which sometimes causes delay, loss of hardcopy documents, and difficulty in following up specific break-fix issues. This manual process appears to be inefficient for maintaining and tracking critical incident database.

Until now, information technology has become a key component to facilitate faster access to critical repair data, with increased accuracy and more convenience in terms of managing the entire workflow.

From that note, a system program and database development for building maintenance has been initiated, as an alternative tool for Building Resource Manager. The program is bundled with systematic collection of information, suitable for accurately tracking of all incident data. By utilizing this computer program, we can ensure proper use of building infrastructure, and more importantly, enable longer life cycle for all building equipment. Lastly, the system can help identify, detect, and effectively prevent future accidents or critical issues from the deterioration of old and obsolete equipment in the building.

Additionally, the use of building maintenance programs will help reduce human errors from the manual process as it can generate status reports periodically or on as-need basis. The reports can ultimately be used for analysis and budget planning purposes.

## กิตติกรรมประกาศ

วิทยานิพนธ์และการพัฒนาโปรแกรมในครั้งนี้สำเร็จลุล่วงได้ เพราะได้รับความกรุณาเป็นอย่างยิ่งจาก ศ.ฐิติพัฒน์ ประทานทรัพย์ ที่ปรึกษาวิทยานิพนธ์ที่ได้ให้โอกาสและคำแนะนำรวมทั้งคำสั่งสอนต่างๆ เพื่อใช้ในการพัฒนาโปรแกรม รวมถึง ผศ.ดร.ปรีชญา มัทชนทวิ อาจารย์ชนะเลิศ อินทรเกษตร อาจารย์ประยุทธ พันธุลาก ที่ได้สั่งสอนและให้ความรู้ต่างๆ ในระหว่างการศึกษา

ขอขอบคุณประธานกรรมการและคณะกรรมการรวมทั้งผู้ทรงคุณวุฒิภายนอก ที่กรุณาเสียสละเวลาและให้คำแนะนำต่างๆ ซึ่งทำให้วิทยานิพนธ์มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบคุณ ฝ่ายงานช่าง มหาวิทยาลัยศิลปากรวิทยาเขตสารสนเทศเพชรบุรี ที่ให้ความอนุเคราะห์ในการให้ข้อมูลและทดสอบโปรแกรม แผนกวิศวกรรมบริการ รพ.นครชน และงานบริการซ่อมบำรุงบริษัท กสท โทรคมนาคมจำกัด (มหาชน) เขตบางรัก

ขอบคุณสมาชิกในครอบครัวทุกท่านที่ช่วยสนับสนุน เป็นกำลังใจและสนับสนุนในทุกๆ ด้านทำให้การศึกษาสำเร็จลุล่วง

ผู้จัดทำขอขอบพระคุณทุกๆ ท่านเป็นอย่างสูง ณ ที่นี้ที่ได้ช่วยให้การจัดทำวิทยานิพนธ์ในครั้งนี้สำเร็จลุล่วงไปได้ด้วยดีคุณค่าหรือประโยชน์อันเกินจากวิทยานิพนธ์ฉบับนี้ข้าพเจ้าขอบแต่พระคุณบิดามารดาครูอาจารย์ที่ได้อบรมสั่งสอนแนะนำสนับสนุนและให้กำลังใจเสมอมา

มาย ยินชัย



## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ง
บทคัดย่อภาษาอังกฤษ .....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง .....	ฎ
สารบัญภาพ .....	ฐ
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	3
1.3 สมมติฐานของการศึกษา.....	3
1.4 ขอบเขตของการศึกษา .....	3
1.5 ขั้นตอนของการศึกษา.....	3
1.6 วิธีการศึกษา.....	3
1.7 ประโยชน์ที่ได้รับจากการศึกษา.....	3
บทที่ 2 แนวความคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง .....	5
2.1 ทฤษฎีอาคารและสภาพแวดล้อม .....	6
2.2 ความเป็นมาและแผนภาพภายใต้ความสัมพันธ์ของ 3 องค์ประกอบ .....	6
2.2.1 ความหมายของ Facility management .....	6
2.3 การบำรุงรักษาและการปรับปรุงอาคารต่างๆ .....	8
2.4 รูปแบบการบริการจัดการซ่อมบำรุงอาคาร .....	10
2.4.1 ผู้บริหารอาคาร.....	10



2.4.2	เจ้าของอาคารเป็นผู้บริหารจัดการ.....	10
2.4.3	เจ้าของอาคารจ้างนิติบุคคลภายนอกหรือบุคคลภายนอกเป็นผู้บริหารจัดการ.....	10
2.5	ลักษณะการจัดการงานบำรุงรักษา.....	11
2.5.1.	ลักษณะของการจัดการบำรุงรักษาที่ไม่ดี.....	11
2.5.2	ลักษณะของการบำรุงรักษาที่ดี.....	12
2.5.3	ปัญหาของระบบการจัดการงานบำรุงรักษาแบบใช้กระดาษ.....	12
2.6	ระบบการจัดการงานบำรุงรักษาอุปกรณ์โดยใช้โปรแกรมคอมพิวเตอร์.....	13
2.6.1	ความสามารถของ CMMS.....	14
2.6.2	ข้อพิจารณาก่อนการนำระบบ CMMS มาใช้.....	15
2.7	การสำรวจลักษณะขั้นตอนการบริหารจัดการอาคารสถานที่.....	16
2.8	แนวคิดและทฤษฎีฐานข้อมูล (Database).....	24
2.8.1	ฐานข้อมูลแบบแตกสาขา (Hierarchical Database).....	24
2.8.2	ฐานข้อมูล แบบเครือข่าย (Network Database).....	25
2.8.3	ระบบฐานข้อมูลแบบสัมพันธ์ (Relation Database).....	26
2.8.4	คุณสมบัติของระบบฐานข้อมูลแบบสัมพันธ์.....	27
2.8.5	องค์ประกอบของระบบฐานข้อมูล (Database System).....	29
2.8.6	คำศัพท์ที่ควรรู้จักเกี่ยวกับฐานข้อมูล.....	30
2.8.7	รูปแบบความสัมพันธ์แบบต่างๆ.....	31
2.9	การศึกษาโปรแกรมที่เกี่ยวข้อง.....	32
2.9.1	ขั้นตอนการทำงานของโปรแกรม CAFM Explorer WorkOrder.....	33
2.9.2	ขั้นตอนการทำงานของโปรแกรม HIPPO.....	38
2.9.3	สรุปการศึกษาโปรแกรมในกลุ่ม Work Order ที่มีอยู่ในปัจจุบัน.....	43
2.9.4	สรุปข้อดีและการนำมาประยุกต์ใช้.....	43
บทที่ 3	กระบวนการพัฒนาโปรแกรม.....	45

3.1	ศึกษาการทำงานของหน่วยงานซ่อมบำรุงรักษาอาคาร .....	46
3.1.1	การเก็บข้อมูลจากการสัมภาษณ์.....	46
3.2	วิเคราะห์ข้อมูลจากฝ่ายซ่อมบำรุงรักษาประจำอาคาร .....	46
3.2.1	ลักษณะงานซ่อมบำรุง .....	46
3.2.2	ขั้นตอนการทำงาน.....	46
3.2.3	ระบบงานแจ้งซ่อมในระบบเดิม .....	48
3.2.4	ปัญหาที่พบในกระบวนการของงานซ่อมบำรุง.....	49
3.2.4.1	ระบบแจ้งซ่อมขาดประสิทธิภาพ .....	49
3.2.4.2	ขาดระบบจัดการงานแจ้งซ่อม.....	49
3.3	การออกแบบระบบงานใหม่ .....	49
3.4	ลักษณะและความสามารถของโปรแกรม .....	51
3.4.1	ส่วนงานแจ้งซ่อม .....	51
3.4.2	ส่วนงานแผนบำรุงรักษาแบบป้องกัน.....	51
3.4.3	ส่วนผู้ใช้งาน.....	51
3.4.4	ส่วนจัดเก็บข้อมูลอุปกรณ์.....	51
3.5	การออกแบบและพัฒนาโปรแกรม .....	52
3.5.1	ส่วนแสดงข้อมูลรายละเอียดของงานประจำวัน.....	52
3.5.2	ส่วนข้อมูลบุคคลที่ทำงานซ่อมบำรุง(ช่าง).....	52
3.5.3	ส่วนข้อมูลอุปกรณ์ประจำอาคาร .....	53
3.5.4	ส่วนข้อมูลงานแจ้งซ่อม .....	53
3.5.5	ส่วนแผนงานซ่อมบำรุง .....	54
3.5.6	ส่วนหน้าจอผู้แจ้งซ่อม.....	54
3.5.7	ส่วนหน้าจอช่าง .....	55
3.6	ขั้นตอนการทำงาน (Project Flow).....	56

3.6.1	ขั้นตอนการทำงานแจ้งซ่อม.....	56
3.6.2	ขั้นตอนการทำงานแผนบำรุงรักษา.....	57
3.7	การออกแบบฐานข้อมูล.....	58
3.7.1	ตารางฐานข้อมูล.....	58
3.7.1.1	กลุ่มฐานข้อมูลบุคคล.....	58
3.7.1.2	กลุ่มฐานข้อมูลการแจ้งซ่อม.....	59
3.7.1.3	กลุ่มฐานข้อมูลแผนบำรุงรักษา.....	61
3.7.2	ความสัมพันธ์ของข้อมูล (ER Diagram)ฐานข้อมูลระบบโปรแกรมบริหารจัดการ บำรุงรักษาอาคาร.....	62
บทที่ 4	ผลการศึกษา.....	64
4.1	โปรแกรมที่ใช้ในการพัฒนาระบบ.....	64
4.2	ผังโครงสร้างหน้าจอโปรแกรม.....	64
4.3	ความสามารถของโปรแกรมในการกำหนดสิทธิ์.....	66
4.4	หน้าจอส่วนติดต่อกับผู้ใช้งาน โปรแกรม (User Interface).....	67
4.4.1	หน้าจอเริ่มต้นโปรแกรมหน้า Login.....	67
4.4.2	หน้าจอลงทะเบียน(SignUp).....	67
4.4.3	หน้าจอ รายงานรายละเอียดของงาน (Dash Board).....	68
4.4.4	หน้าจอเมนูของโปรแกรม.....	69
4.4.5	หน้าจอเพิ่มข้อมูลช่าง.....	70
4.4.6	หน้าจอข้อมูลรายละเอียดของช่าง.....	70
4.4.7	หน้าจอเพิ่มอุปกรณ์.....	71
4.4.8	หน้าจอรายละเอียดอุปกรณ์.....	73
4.4.9	หน้าจอแจ้งซ่อม.....	74
4.4.10	หน้าจอมอบหมายงานซ่อม.....	75

4.4.11 หน้าจอตรวจสอบรายการแจ้งซ่อม .....	76
4.4.12 เพิ่มแผนซ่อมบำรุง(กำหนดตัวอุปกรณ์).....	77
4.4.13 หน้าจอสร้างแผนซ่อมบำรุง.....	78
4.4.14 หน้าจอตรวจสอบแผนการบำรุงรักษา .....	79
4.4.15 หน้าจอแสดงงานที่ได้รับมอบหมายของช่าง .....	80
4.4.16 หน้าจอผู้แจ้งซ่อม .....	80
4.4.17 หน้าจอรายงานรวม.....	82
บทที่ 5 สรุปผลการศึกษาและข้อเสนอแนะ.....	86
5.1 สรุปผลการพัฒนาโปรแกรมซ่อมบำรุงรักษาอาคาร.....	86
5.1.1 สรุปความสามารถของโปรแกรม.....	86
5.2 ปัญหาและอุปสรรค .....	87
5.3 ข้อเสนอแนะ .....	87
ภาคผนวก ชุดคำสั่งที่ใช้พัฒนาโปรแกรม.....	88
1.ชุดคำสั่งหน้าจอ Login.aspx.....	89
2. ชุดคำสั่งหน้า DashBoard.aspx.cs .....	91
3. ชุดคำสั่งหน้า AddPlanMaintenance.aspx.cs.....	95
4.ชุดคำสั่งหน้า ReportMaintenanceJob.aspx.cs .....	97
5.ชุดคำสั่งสร้าง QR_Code อ้างอิงไฟล์ AddEquipment.aspx.cs .....	100
6.ชุดคำสั่ง Database.cs .....	100
7.ชุดคำสั่ง SqlDb.cs.....	117
รายการอ้างอิง .....	122
ประวัติผู้เขียน .....	125

## สารบัญตาราง

	หน้า
ตารางที่ 1 ส่วนหัวข้อแสดงข้อมูลรายละเอียดของงานประจำวัน .....	52
ตารางที่ 2 แสดงหัวข้อส่วนข้อมูลบุคคลที่ทำงานซ่อมบำรุง(ช่าง) .....	52
ตารางที่ 3 แสดงหัวข้อส่วนข้อมูลอุปกรณ์ประจำอาคาร.....	53
ตารางที่ 4 แสดงหัวข้อส่วนข้อมูลงานแจ้งซ่อม .....	53
ตารางที่ 5 แสดงหัวข้อส่วนแผนงานซ่อมบำรุง .....	54
ตารางที่ 6 แสดงหัวข้อส่วนหน้าจอผู้แจ้งซ่อม .....	54
ตารางที่ 7 แสดงหัวข้อส่วนหน้าจอช่าง .....	55
ตารางที่ 8 แสดงตารางข้อมูลชื่อ UserInfo .....	58
ตารางที่ 9 แสดงตารางข้อมูลชื่อ Ref_Position .....	59
ตารางที่ 10 แสดงตารางข้อมูลชื่อ TechnicianProfile.....	59
ตารางที่ 11 แสดงตารางข้อมูลชื่อ MaintenanceStatus.....	59
ตารางที่ 12 แสดงตารางข้อมูลชื่อ DefectedStuff.....	60
ตารางที่ 13 แสดงตารางข้อมูลชื่อ MaintenanceJob .....	60
ตารางที่ 14 แสดงตารางข้อมูลชื่อ EquipmentDetail .....	61
ตารางที่ 15 แสดงตารางข้อมูลชื่อ MaintenancePlanDetail.....	61
ตารางที่ 16 แสดงรายละเอียดของสิทธิในกลุ่มผู้ใช้งานต่างๆในระบบ.....	66

## สารบัญภาพ

หน้า

ภาพที่ 1	แผนภาพแสดงองค์ประกอบของการบริหารจัดการทรัพยากรอาคาร.....	8
ภาพที่ 2	ภาพรวมของขอบเขตในการปฏิบัติการจัดการอาคาร .....	9
ภาพที่ 3	แสดง โปรแกรมของระบบ CMMS บนจอคอมพิวเตอร์ [8].....	15
ภาพที่ 4	แสดงความสัมพันธ์ของสายงาน บริษัท กสท. โทรคมนาคมจำกัด (มหาชน) เขตบางรัก .18	
ภาพที่ 5	แสดงขั้นตอนการดำเนินงานซ่อมบำรุง บริษัท กสท. โทรคมนาคมจำกัด (มหาชน) เขต บางรัก.....	19
ภาพที่ 6	แผนผังการบังคับบัญชาแผนกวิศวกรรมบริการ รพ.นครธน .....	20
ภาพที่ 7	ขั้นตอนการทำงานแผนกวิศวกรรมบริการ รพ.นครธน .....	21
ภาพที่ 8	ความสัมพันธ์ของฝ่ายงานช่าง มหาวิทยาลัยศิลปากร วิทยาเขตสารสนเทศเพชรบุรี .....	22
ภาพที่ 9	ขั้นตอนการทำงานของฝ่ายงานช่าง มหาวิทยาลัยศิลปากร วิทยาเขตสารสนเทศเพชรบุรี ..23	
ภาพที่ 10	ตัวอย่างฐานข้อมูลแบบแตกสาขา .....	24
ภาพที่ 11	ฐานข้อมูลแบบเครือข่าย .....	25
ภาพที่ 12	รูปตัวอย่างการเก็บข้อมูลแบบตาราง.....	26
ภาพที่ 13	แสดงความสัมพันธ์ 1 : 1 .....	31
ภาพที่ 14	แสดงความสัมพันธ์ 1 : M.....	31
ภาพที่ 15	แสดงความสัมพันธ์แบบ M : N .....	32
ภาพที่ 16	แสดงหน้าจอหลักของโปรแกรม [10] .....	33
ภาพที่ 17	แสดงหน้าจอการแจ้ง WorkOrder [10].....	34
ภาพที่ 18	แสดงงาน WorkOrder ที่ทำการแจ้งเข้ามาในหน้าจอโปรแกรมหลัก [10].....	34
ภาพที่ 19	แสดงการเลือกผู้รับผิดชอบงาน [10].....	35
ภาพที่ 20	แสดงสถานะของงานเปลี่ยนเป็น Open [10].....	35

ภาพที่ 21 ผู้ได้รับมอบหมายงานทำการ Login เข้าสู่ระบบ [10] .....	36
ภาพที่ 22 ส่วนแสดงผลสำหรับช่างที่ได้รับมอบหมายงาน [10] .....	36
ภาพที่ 23 แสดงส่วนเริ่มดำเนินการซ่อมช่างยืนยันเริ่มการทำงาน [10].....	37
ภาพที่ 24 หน้าจอหลักของโปรแกรม HIPPO [12] .....	38
ภาพที่ 25 หน้าจอแสดงการรับงานแจ้งซ่อม โปรแกรม HIPPO [12] .....	39
ภาพที่ 26 แสดงการค้นหารายการแจ้งซ่อมที่มีเข้ามาในโปรแกรม [12] .....	40
ภาพที่ 27 แสดงรายละเอียดลักษณะงานแจ้งซ่อม [12] .....	41
ภาพที่ 28 แสดงรายละเอียดส่วนรายงานผลสำหรับเลือกเพื่อสร้างรายงานการแจ้งซ่อม [12].....	41
ภาพที่ 29 แสดงการกรอกข้อมูลการแจ้งซ่อมในหน้าจอหลัก [10] .....	44
ภาพที่ 30 หน้าจอแสดงข้อมูลในรูปแบบตาราง [10] .....	44
ภาพที่ 31 แสดงการใช้สีในการสื่อถึงความแตกต่างของข้อมูล [12] .....	44
ภาพที่ 32 แผนภูมิแสดงวิธีดำเนินการศึกษา .....	45
ภาพที่ 33 แผนภูมิแสดงขั้นตอนงานซ่อมบำรุงตามแผนการซ่อมบำรุงที่มีการกำหนดไว้ล่วงหน้า..	47
ภาพที่ 34 แผนภูมิแสดงขั้นตอนงานซ่อมบำรุงที่มีการร้องขอหรือแจ้งซ่อมเข้ามา .....	47
ภาพที่ 35 แสดงระบบการทำงานแจ้งซ่อมในระบบเดิม.....	48
ภาพที่ 36 แสดงระบบการทำงานที่ออกแบบใหม่ .....	50
ภาพที่ 37 Flow Chart แสดงขั้นตอนการทำงานแจ้งซ่อม.....	56
ภาพที่ 38 Flow Chart แสดงขั้นตอนการทำงานบำรุงรักษา .....	57
ภาพที่ 39 ฐานข้อมูลระบบโปรแกรมบริหารจัดการบำรุงรักษาอาคาร (ER Diagram).....	63
ภาพที่ 40 แสดงแผนภูมิการเชื่อมโยงหน้าเว็บในโปรแกรม .....	65
ภาพที่ 41 แสดงรูปหน้าจอการเข้าใช้โปรแกรม .....	67
ภาพที่ 42 แสดงหน้าจอ การลงทะเบียนสมัครสมาชิกเพื่อเข้าใช้งานการแจ้งซ่อม .....	67
ภาพที่ 43 แสดงหน้าจอ รายงานรายละเอียดของงาน (DashBoard) .....	68
ภาพที่ 44 แสดงหน้าจองานแจ้งซ่อมประจำวัน.....	68



ภาพที่ 45 แสดงหน้าจองานแผนซ่อมบำรุงประจำวัน .....	69
ภาพที่ 46 แสดงหน้าจอเมนูของ โปรแกรม .....	69
ภาพที่ 47 แสดงหน้าจอเพิ่มข้อมูลช่าง .....	70
ภาพที่ 48 แสดงหน้าจอข้อมูลรายละเอียดของช่าง .....	71
ภาพที่ 49 แสดงหน้าจอข้อมูลรายละเอียดของช่างที่ส่งออกไปยัง โปรแกรม Excel .....	71
ภาพที่ 50 แสดงหน้าจอเพิ่มข้อมูลอุปกรณ์และแสดงรหัส QR Code .....	72
ภาพที่ 51 แสดงหน้าจอรายละเอียดอุปกรณ์ .....	73
ภาพที่ 52 แสดงหน้าจองานแจ้งซ่อม .....	74
ภาพที่ 53 แสดงหน้าจอมอบหมายงานซ่อม .....	75
ภาพที่ 54 แสดงหน้าจอตรวจสอบรายการแจ้งซ่อม .....	76
ภาพที่ 55 แสดงข้อมูลที่ส่งไปยังโปรแกรม Excel .....	76
ภาพที่ 56 แสดงหน้าจอค้นหาอุปกรณ์ที่จะนำไปสร้างแผนการซ่อมบำรุง .....	77
ภาพที่ 57 แสดงหน้าจอสร้างแผนซ่อมบำรุง .....	78
ภาพที่ 58 แสดงหน้าจอตรวจสอบแผนการบำรุงรักษา .....	79
ภาพที่ 59 แสดงข้อมูลที่ส่งไปยังโปรแกรม Excel .....	79
ภาพที่ 60 แสดงหน้าจองานที่ได้รับมอบหมายของช่าง .....	80
ภาพที่ 61 แสดงหน้าจอหลักของผู้แจ้งซ่อม .....	80
ภาพที่ 62 หน้าต่างแจ้งซ่อมสำหรับผู้แจ้งซ่อม .....	81
ภาพที่ 63 รายงานการซ่อมประจำวัน .....	82
ภาพที่ 64 รายงานการซ่อมประจำเดือน .....	82
ภาพที่ 65 รายงานแผนบำรุงรักษาประจำวัน .....	83
ภาพที่ 66 รายงานแผนบำรุงรักษาประจำเดือน .....	83
ภาพที่ 67 รายงานจำนวนการแจ้งซ่อมแยกตามประเภทอุปกรณ์ .....	84
ภาพที่ 68 รายงานจำนวนการซ่อมต่ออุปกรณ์ .....	84



ภาพที่ 69 รายงานเวลาในการทำงานของช่าง .....85

ภาพที่ 70 รายงานค่าใช้จ่ายในการบำรุงรักษา.....85

ภาพที่ 71 หน้าจอ Login .....89

ภาพที่ 72 หน้าจอ DashBoard.....91

ภาพที่ 73 หน้าจอ AddPlanMaintenance .....95

ภาพที่ 74 ภาพหน้าจอตรวจสอบรายการแจ้งซ่อม .....97



## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

การบริหารทรัพยากรกายภาพ (FM : Facility Management) มีเนื้อหาครอบคลุมการจัดการทุกกิจกรรมในองค์กรที่เกี่ยวข้องรวมไปถึงการจัดการผลกระทบของสิ่งแวดล้อมที่มีผลต่อบุคลากรและสถานที่ทำงาน เช่น กำกับการใช้พื้นที่ อุปกรณ์อาคาร อุปกรณ์สำนักงาน ดูแลซ่อมบำรุงอาคาร สิ่งก่อสร้าง สถานที่และสภาพแวดล้อม ให้มีความพร้อมและตอบสนองการใช้งาน เพื่อให้เกิดประโยชน์สูงสุดต่อผู้ใช้และเจ้าของ โดยกำหนดให้ตัวอาคารเป็นเครื่องมือที่คอยสนับสนุนองค์กร ในการเพิ่มประสิทธิภาพและประสิทธิผลขององค์กร ทั้งในด้านการเพิ่มผลผลิต การลดค่าใช้จ่ายในการปฏิบัติงานในระยะยาว สามารถเห็นปัจจัยของต้นทุนที่แท้จริง ทำให้ผู้บริหารการเงิน ผู้บริหารทรัพยากรสินค้านำมาพิจารณาตัดสินใจได้อย่างแม่นยำซึ่งแนวความคิดในการบริหารทรัพยากรกายภาพจะมุ่งเน้นในการให้บริการ ต่อผู้คนในอาคาร (People) การทำงาน (Process) และอาคารสถานที่ (Place) ให้สามารถทำงานกันได้อย่างสอดคล้องเพื่อบรรลุผลสำเร็จขององค์กรตามที่มุ่งหมายไว้

งานดูแลอาคาร (Building Maintenance) การจัดการภายหลังการเข้าใช้งานอาคารในส่วนของงานดูแลอาคาร ผู้บริหารจัดการทรัพยากรอาคารมีหน้าที่ในการตรวจสอบสภาพการใช้งานของสาธารณูปโภคต่าง ๆ ในอาคาร ทุกงานเพื่อให้ระบบทำงานได้อย่างมีประสิทธิภาพ ซึ่งโดยทั่วไปอาคารต่างๆ จะมีหน่วยซ่อมบำรุงประจำอาคารเพื่อคอยแก้ไขปัญหาต่างๆ ที่จะเกิดขึ้น การซ่อมบำรุงอาคาร จะใช้กระบวนการบันทึกข้อมูลต่างลงกระดาษแบบฟอร์มหรือการติดต่อผ่านโทรศัพท์เพื่อแจ้งปัญหาที่เกิดขึ้น กระบวนการลักษณะนี้มักเกิดปัญหาล่าช้าในการส่งเอกสาร เอกสารสูญหาย มีความยากลำบากในการติดตามความคืบหน้าของงาน และยังขาดประสิทธิภาพในการรวบรวมข้อมูลการซ่อมต่างๆ

เนื่องจากในปัจจุบันเทคโนโลยีได้เข้ามามีบทบาทต่อการทำงานอย่างมากซึ่งก็สามารถช่วยให้การทำงานต่างๆรวดเร็ว สะดวก ถูกต้อง มากยิ่งขึ้น โดยได้มีการนำเทคโนโลยีมารวมกับแนวความคิดในการบริหารทรัพยากรกายภาพ ซึ่งเป็นที่มาของคอมพิวเตอร์ช่วยบริหาร

ทรัพยากรกายภาพ (CAFM : Computer Aided Facility Management) [1] ซึ่งระบบนี้ถูกออกแบบมาเพื่อช่วยให้การจัดการข้อมูล และการจัดการระบบการทำงานต่างๆ โดยโปรแกรมสำเร็จรูปที่วางจำหน่ายในท้องตลาดส่วนใหญ่จะมีหน้าที่ต่างๆดังนี้

- การบริหารพื้นที่การบริหารอุปกรณ์สำนักงานและเฟอร์นิเจอร์
- การบริการพื้นที่เช่า
- การบริหารการบำรุงรักษา
- การบริหารระบบสาธารณูปโภค
- การบริหารการเงิน
- การบริหารการอพยพออกจากอาคาร
- อื่นๆ

และด้วยฟังก์ชันการทำงานจำนวนมากเหล่านี้จึงทำให้ราคาของโปรแกรมสูงตามไปด้วย ถึงแม้จะมีการจำหน่ายโปรแกรมแยกเป็น Module เฉพาะงานแต่ก็ยังมีราคาสูง จึงเป็นเรื่องยากที่ผู้บริหารอาคาร จะลงทุนนำมาใช้ หรือถ้าผู้บริหารต้องการโปรแกรม สำหรับบริหารงานเฉพาะเรื่อง เช่น หากต้องการ โปรแกรมสำหรับบริหารการบำรุงรักษาส่วนของอาคารอย่างเดียว การลงทุนซื้อชุดโปรแกรมสำเร็จรูปจากต่างประเทศอาจเป็นการลงทุนที่ไม่คุ้มค่านี้อาจมีฟังก์ชันการทำงานที่ไม่สอดคล้องกับระบบงานภายในองค์กร และจากการสำรวจและศึกษาโปรแกรมสำเร็จรูปที่มีวางจำหน่ายในปัจจุบันเช่น โปรแกรม ManageEngine\_FacilitiesDesk พบว่าฟังก์ชันการทำงานบางส่วนมีความซับซ้อนและเหมาะกับองค์กรขนาดใหญ่ที่มีสาขาหลายแห่ง

จึงเป็นที่มาของการพัฒนาโปรแกรมสำหรับบำรุงรักษาซ่อมแซมอาคาร เพื่อเป็นทางเลือกสำหรับองค์กรที่มีอาคารสถานที่ และมีการเก็บข้อมูลที่เหมาะสมกับลักษณะงานตามโครงสร้างขององค์กร ซึ่งการจัดการอย่างเป็นระบบผ่านโปรแกรมคอมพิวเตอร์จะช่วยให้ ระบบสาธารณูปโภคภายในอาคารมีอายุการใช้งานที่ยาวนานขึ้นและยังเป็นการป้องกันอุบัติเหตุที่อาจเกิดขึ้นจากความเสื่อมของอุปกรณ์ภายในอาคาร อีกทั้งการนำโปรแกรมบริหารจัดการบำรุงรักษาอาคารมาใช้จะช่วยลดข้อผิดพลาดต่างๆ ที่จะเกิดจากผู้ปฏิบัติงานได้ นอกจากนี้ยังสามารถรายงานสถานะต่างๆ ที่นำไปใช้ในการวิเคราะห์และวางแผนการทำงาน การตั้งงบประมาณในอนาคตได้อีกด้วย

## 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. ศึกษารูปแบบการบริหารจัดการซ่อมบำรุงอาคาร การใช้คอมพิวเตอร์ช่วยในการบำรุงรักษาอาคาร

2. เพื่อออกแบบและพัฒนาโปรแกรมสำหรับจัดการงานบำรุงรักษาอุปกรณ์ภายในอาคาร ซึ่งสามารถเก็บรวบรวมข้อมูลต่างๆในการซ่อมบำรุงแต่ละครั้งได้

## 1.3 สมมติฐานของการศึกษา

โปรแกรมสามารถรองรับการทำงานซ่อมแซมระบบและบำรุงรักษาสาธารณูปโภคภายในอาคาร สามารถเก็บรวบรวมข้อมูลอุปกรณ์อาคารที่จะต้องบำรุงรักษาและ ติดตามผลการทำงาน มีการรายงานผลสถานะความคืบหน้าในกระบวนการซ่อมบำรุงได้

## 1.4 ขอบเขตของการศึกษา

1. พัฒนาโปรแกรมเพื่อใช้ในการจัดการงานซ่อมบำรุง สำหรับอาคารสำนักงานที่ไม่มีระบบบริหารอาคารอัตโนมัติ (Building Automation System)

## 1.5 ขั้นตอนของการศึกษา

- เก็บข้อมูลขั้นตอนการทำงานบำรุงรักษาซ่อมแซมอาคาร
- วิเคราะห์ขั้นตอนหรือลักษณะการทำงาน
- ออกแบบฐานข้อมูลของโปรแกรม
- พัฒนาโปรแกรม
- ทดลองและประเมินผลการทำงานของโปรแกรม

## 1.6 วิธีการศึกษา

เก็บข้อมูลและศึกษาขั้นตอนการทำงานจากหน่วยงานที่มีหน้าที่บำรุงซ่อมแซมอาคาร และนำมาวิเคราะห์เพื่อพัฒนาโปรแกรมเข้ามาช่วยในการทำงาน

## 1.7 ประโยชน์ที่ได้รับจากการศึกษา

ประโยชน์ที่ได้รับจากการพัฒนาโปรแกรมนี้นี้

ได้โปรแกรมที่สามารถใช้งานอย่างเหมาะสมกับหน่วยงานซ่อมบำรุงรักษาอาคาร  
ขนาด เล็ก – กลาง และยังสามารถเก็บรวบรวมข้อมูลต่างๆ ในการซ่อมบำรุงแต่ละครั้งได้ เพื่อช่วยให้  
ผู้จัดการอาคารหรือผู้ดูแลอาคารมีข้อมูลต่างๆ ไว้พิจารณาหรือตัดสินใจ ดำเนินการต่างๆ เพื่อให้การ  
ซ่อมบำรุงมีประสิทธิภาพมากขึ้น ซึ่งก็จะส่งผลให้การใช้งานอาคารมีประสิทธิภาพมากขึ้นตามไป  
ด้วย



## บทที่ 2

### แนวความคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในบทนี้เป็นการรวบรวมและศึกษาข้อมูลแนวคิดและทฤษฎีที่เกี่ยวข้องในการพัฒนาระบบโปรแกรมบริหารจัดการบำรุงรักษาอาคาร เพื่อให้ทราบแนวทางการพัฒนาโปรแกรม และ ศึกษา ลักษณะกระบวนการขั้นตอนการบริหารจัดการทรัพยากรอาคาร งานซ่อมบำรุงรักษาอาคาร แนวคิด การเก็บข้อมูลระบบการแจ้งซ่อมในแต่ละงาน และการบริหารงานซ่อมบำรุง เพื่อให้สอดคล้องกับการพัฒนาโปรแกรมและตรงกับความต้องการของผู้ใช้งาน โดยขอบเขตหัวข้อที่ทำการศึกษามีดังนี้

- 2.1. ทฤษฎีอาคารและสภาพแวดล้อม
- 2.2 ความเป็นมา
  - 2.2.1 ความหมายของ Facility management
- 2.3 การบำรุงรักษาและการปรับปรุงอาคารต่างๆ
- 2.4 รูปแบบการบริการจัดการซ่อมบำรุงอาคาร
  - 2.4.1 ผู้บริหารอาคาร
  - 2.4.2 เจ้าของอาคารเป็นผู้บริหารจัดการ
  - 2.4.3 เจ้าของอาคารจ้างนิติบุคคลภายนอกหรือบุคคลภายนอกเป็นผู้บริหารจัดการ
- 2.5 ลักษณะการจัดการงานบำรุงรักษา
  - 2.5.1 ลักษณะของการจัดการบำรุงรักษาที่ไม่ดีมักจะประกอบด้วย
  - 2.5.2 ลักษณะของการบำรุงรักษาที่ดีมักประกอบด้วย
  - 2.5.3 ปัญหาของระบบการจัดการงานบำรุงรักษาแบบใช้กระดาษ
- 2.6 ระบบการจัดการงานบำรุงรักษาอุปกรณ์โดยใช้โปรแกรมคอมพิวเตอร์
  - 2.6.1 ความสามารถของ CMMS
  - 2.6.2 ข้อพิจารณาก่อนการนำระบบ CMMS มาใช้
- 2.7 การสำรวจลักษณะขั้นตอนการบริหารจัดการอาคารสถานที่
- 2.8 แนวคิดและทฤษฎีฐานข้อมูล (Database)
- 2.9 ศึกษาโปรแกรมที่เกี่ยวข้อง

## 2.1 ทฤษฎีอาคารและสภาพแวดล้อม

อาคารสถานที่ที่ไม่ได้หมายถึงในแง่กายภาพเพียงอย่างเดียว หรือเฉพาะบริเวณที่เป็นพื้นที่อาคาร หรือสิ่งปลูกสร้างแต่ยังหมายถึง สิ่งที่สนับสนุนก่อให้เกิดผลงาน ผลผลิต และรายได้ การใช้อาคารสถานที่จะต้องมีความเข้าใจ ถึงองค์ประกอบด้านกายภาพ ลักษณะเฉพาะและข้อจำกัดต่างๆ เพื่อที่จะสามารถใช้อาคารสถานที่ก่อให้เกิดประโยชน์สูงสุดได้[2]

## 2.2 ความเป็นมาและแผนภาพภายใต้ความสัมพันธ์ของ 3 องค์ประกอบ

### 2.2.1 ความหมายของ Facility management

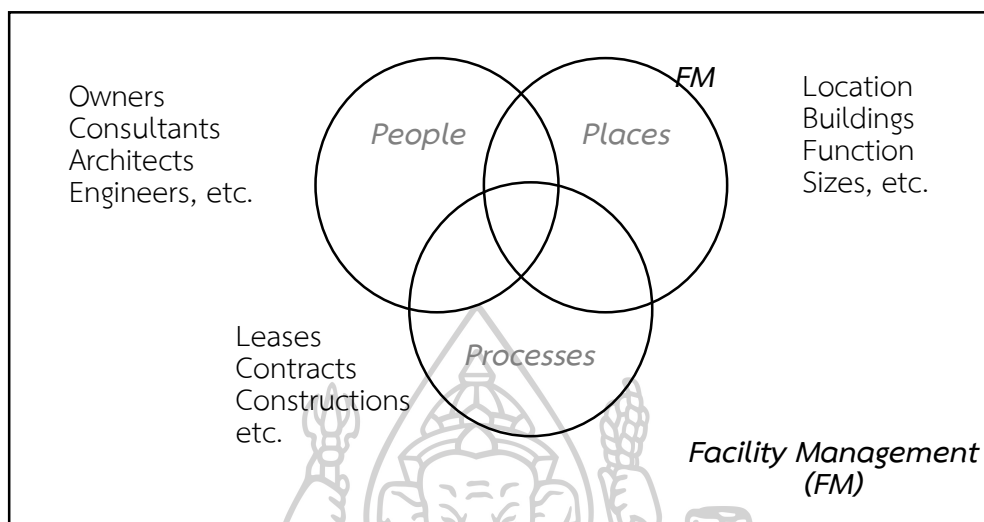
การบริหารจัดการทรัพยากรอาคาร[3] รูปแบบการจัดการสภาพแวดล้อมเพื่ออนาคต การบริหารจัดการทรัพยากรอาคาร (Facility Management, Facilities Management: FM) เป็นวิทยาการที่ได้กำเนิดขึ้น ในช่วงกลางทศวรรษที่ 20 ในประเทศสหรัฐอเมริกา เนื่องมาจากภายหลังการเปลี่ยนผ่านเข้าสู่สังคมยุคอุตสาหกรรม โดยการขยายตัวของระบบการคมนาคมขนส่งนั้น ปริมาณอาคารสูงและการขยายตัวของชุมชนเมืองเพิ่มมากขึ้นอย่างรวดเร็ว ส่งผลให้เกิดความต้องการพื้นที่เพื่อรองรับธุรกิจและกิจกรรมต่าง ๆ โดยเฉพาะในช่วง 50 ปีระหว่าง 1870 - 1920 ซึ่งจำนวนประชากรได้เพิ่มขึ้นเป็น 2 เท่าในขณะที่ความต้องการพื้นที่ดังกล่าวเพิ่มขึ้นถึง 5 เท่า นอกจากนั้นยังมีการก่อตั้งองค์กรขนาดใหญ่ขึ้น อย่างเช่น ในปี 1879 ได้มีการก่อตั้ง Standard Oil อันส่งผลกระทบต่อความต้องการแรงงาน จึงเกิดการอพยพของประชากรเข้ามาอยู่อาศัยและทำงานในเมืองใหญ่เพิ่มมากขึ้น เช่น New York City, Chicago, Tokyo หรือ London ก่อให้เกิดการก่อสร้างอาคารสูงในเมืองใหญ่ เหล่านี้ส่งผลให้ราคาที่ดินสูงขึ้นและทำให้เจ้าของอาคารและที่ดินต้องการหาผลประโยชน์จากทรัพยากร (ที่ดิน) ที่มีอยู่อย่างจำกัดให้มากที่สุด และถูกผลักดันให้มีหน้าที่ในการบริหารจัดการสาธารณูปโภคของอาคาร (ในที่นี้ facility คือ building facility ที่เป็นสาธารณูปโภคของอาคาร)



เพื่อเพิ่มความสามารถในการใช้งานได้ของระบบสาธารณูปโภคของอาคาร ส่งผลให้เกิดสาขาวิชาซีพีการบริหารจัดการอาคาร (building management) ขึ้นมาการบริหารจัดการอาคาร (building management หรือ property management) ในช่วงสมัยแรก ๆ นั้น เป็นลักษณะของงานในการบริหารจัดการที่สัมพันธ์กับงานระบบอาคารเท่านั้น ผู้บริหารระดับสูงยังมิได้เล็งเห็นความสำคัญในแง่ของการประกอบธุรกิจมากนัก จนกระทั่งเกิดวิกฤตการณ์ทางเศรษฐกิจครั้งใหญ่ในปี 1930 ส่งผลกระทบบังให้เกิดการปรับโครงสร้างการบริหารองค์กรโดยรวมในการพยายามลดค่าใช้จ่ายในการปฏิบัติงานลง เกิดกระแสของความสนใจในเรื่องความปลอดภัย (security) พลังงาน (energy) เทคโนโลยี (technology) ผ่านเข้ามาในระบบการบริหารจัดการโดยรวมขององค์กรจนกระทั่งในปี 1970 ได้เกิดวิกฤตการณ์น้ำมันในระดับโลกอย่างรุนแรงและเกิดวิกฤตการณ์ทางเศรษฐกิจอีกครั้ง ทำให้เป็นจุดสำคัญของการกำเนิดแนวทางการบริหารจัดการสมัยใหม่ของ “การบริหารจัดการทรัพยากรอาคาร (Facility Management, Facilities Management หรือ FM)” เข้ามามีส่วนร่วมในการบริหารจัดการเพื่อเข้าถึงเป้าหมายขององค์กร (corporate goal) โดยในการปฏิบัติอาชีพนั้น การบริหารจัดการทรัพยากรอาคาร (facility management) มีบทบาทในการปรับเปลี่ยนและเพิ่มประสิทธิภาพของงานระบบอาคาร การลดค่าใช้จ่าย การสร้าง/วางแผนกลยุทธ์ร่วมกับองค์กร และการบริหารจัดการทรัพยากรอาคารในองค์กรรวม (ในที่นี้ facility คือทรัพยากรอาคารที่เป็น “ทุน” ทุกอย่างที่สามารถทำให้งานหลักขององค์กรดำเนินไปถึงเป้าหมาย) ให้มีประสิทธิภาพมากขึ้นเมื่อพิจารณาความเป็นอยู่ของมนุษย์ในสังคมยุคใหม่ที่มีแนวโน้มที่เป็นไปได้อย่างยิ่งว่าจะต้องอยู่อย่างเข้าใจสิ่งแวดล้อมมากขึ้น ซึ่งต้องประกอบไปด้วยความสมบูรณ์ทั้งทางด้านกายภาพเศรษฐกิจและสังคม การบริหารจัดการ ทรัพยากรและสภาพแวดล้อมในอาคาร (facility and environment management) ได้ถูกพัฒนาขึ้นมาภายใต้ วิกฤตการณ์เศรษฐกิจของโลกที่ตระหนักถึงมูลค่าและความซับซ้อนของอาคาร/สิ่งก่อสร้าง เป็นการบริหารจัดการเชิงบูรณาการในสังคมยุคใหม่ภายใต้หลักการของการสร้างยุทธศาสตร์หรือวางแผนยุทธศาสตร์ โดยคำนึงถึงทรัพยากรมนุษย์เป็นปัจจัยสำคัญเพื่อตอบรับกับโลกที่แปรเปลี่ยนอย่างรวดเร็ว นั่น กล่าวคือ เป็นการจัดการสภาพแวดล้อมอาคารที่ต้องคำนึงถึง มนุษย์ (people) สภาพแวดล้อมสถานที่ (places) และกระบวนการทำงาน (processes) โดยเป็นการจัดการเชิงบูรณาการที่สัมพันธ์กับทฤษฎีและแนวคิดทางสถาปัตยกรรม วิศวกรรม การเงินและการบัญชี การบริหารจัดการ และพฤติกรรมมนุษย์ซึ่งทฤษฎีและแนวคิดเหล่านี้ มีพัฒนาการทั้งทางภาคทฤษฎีและภาคปฏิบัติมาเป็นเวลานาน โดยที่การบริหารจัดการทรัพยากร



อาคาร (facility management) นั้น ได้มีพัฒนาการอยู่บนรากฐานของการพัฒนาทฤษฎีและแนวคิดดังกล่าว เพื่อสร้างสรรค์ทั้งเชิงทฤษฎีและเชิงปฏิบัติงานใหม่ขึ้นมา ดังแสดงในภาพประกอบที่ 1 [3]

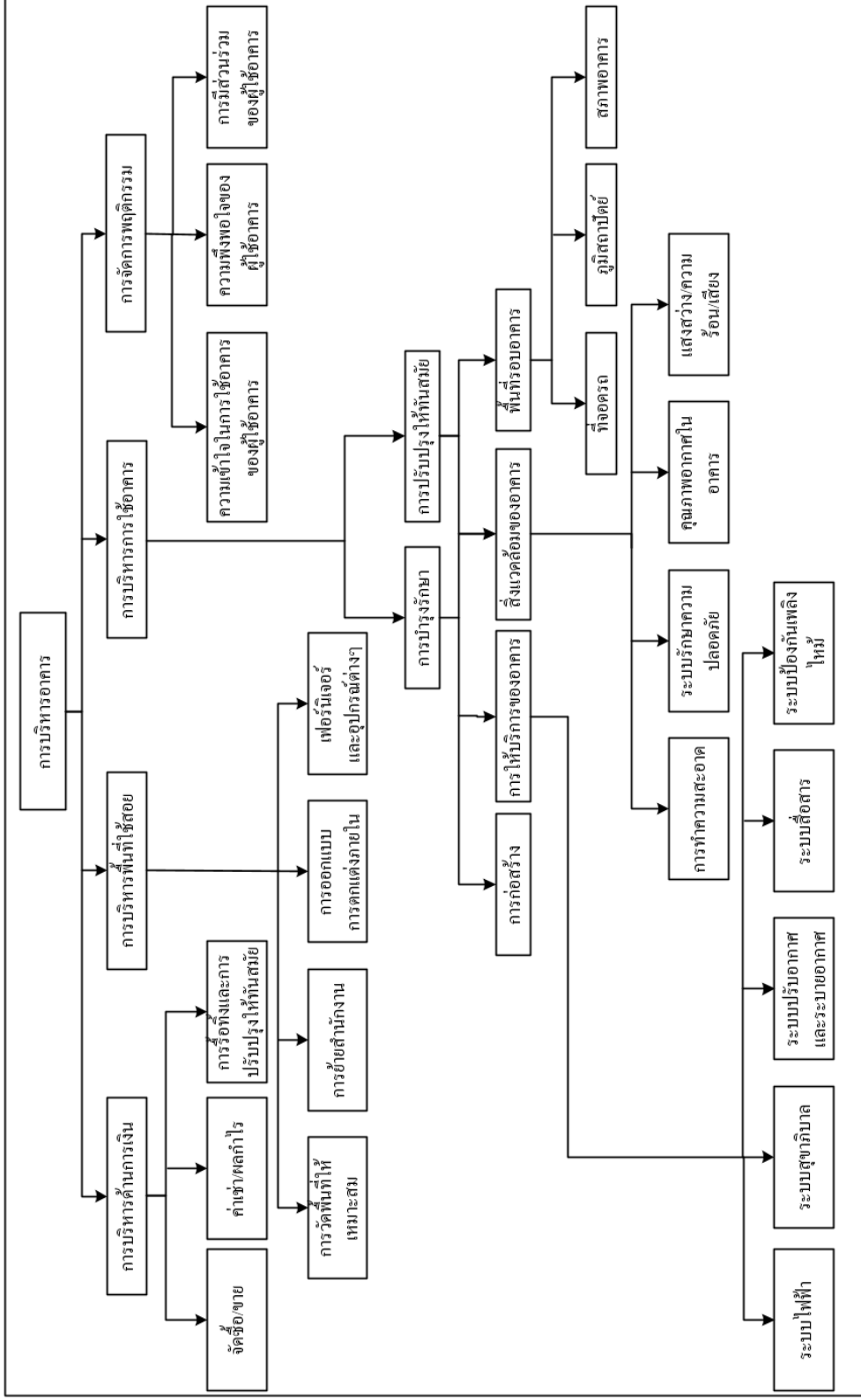


ภาพที่ 1 แผนภาพแสดงองค์ประกอบของการบริหารจัดการทรัพยากรอาคาร

### 2.3 การบำรุงรักษาและการปรับปรุงอาคารต่างๆ

จากที่มีการพัฒนาอย่างรวดเร็วในด้านเทคโนโลยี และความต้องการที่เพิ่มขึ้นของผู้ครอบครองอาคาร ทำให้หน้าที่การบริหารจัดการสิ่งอำนวยความสะดวก มีความเคลื่อนไหวอย่างมาก ทิศทางของการนำการบริหารอาคาร (Facility Management) มาใช้นั้นเริ่มที่จะมีการระบุเจาะจงอย่างชัดเจนในประเทศตะวันตก ทางด้านผู้เชี่ยวชาญและผู้ที่มีส่วนร่วมในการบริหาร ได้แสดงขอบเขตและหน้าที่ของ FM ออกมา โดยแบ่งเป็นกลุ่มตามหัวข้อที่สำคัญตามที่แสดงในรูปที่ 2 [4]

ในหัวข้อต่างๆ ที่แสดงนั้นในด้านการจัดการผู้บริหารก็จะเน้นต่างกันไปตามลักษณะผู้ใช้อาคารและลักษณะของอาคาร เช่น อาคารสำนักงานจะให้ความสำคัญในเรื่องการจัดการเกี่ยวกับการเงินมากกว่า การจัดการด้านความสะดวกสบายในการอยู่อาศัย ในทำนองกลับกัน ที่พักอาศัยจะให้ความสำคัญกับการจัดการเรื่องความสะดวกสบายในการพักอาศัยมากกว่า แต่อย่างไรก็ตามในการจัดการระบบการปฏิบัติการและบำรุงรักษานั้น ก็ยังมีความสำคัญมาก โดยจะทำให้อาคารนั้นมีความสมบูรณ์แบบในด้านประสิทธิภาพ และจะมีผลไปยังค่าใช้จ่ายที่ต้องใช้เพื่อให้เกิดประโยชน์มากที่สุด



ภาพที่ 2 ภาพรวมของขอบเขตในการปฏิบัติการบริหารอาคาร

## 2.4 รูปแบบการบริการจัดการซ่อมบำรุงอาคาร

### 2.4.1 ผู้บริหารอาคาร

การบริหารจัดการอาคาร [5](การจัดการใช้สอยอาคาร) ในประเทศไทยถือเป็นเรื่องที่มีมานานพร้อมๆ กับการเกิดของอาคาร ไม่ว่าจะเป็นอาคารขนาดเล็กหรืออาคารขนาดใหญ่ ต่างก็มีการจัดการด้านบริหารทั้งสิ้น ดังเช่น การทำทะเบียนทรัพย์สิน งานซ่อมแซม ซ่อมบำรุง การเก็บค่าสาธารณูปโภคต่างๆ เป็นต้น งานจัดการเหล่านี้เป็นส่วนหนึ่งของการบริหารจัดการอาคารที่เจ้าของอาคารหรือผู้ใช้อาคารได้ดำเนินไปอย่างเป็นปกติ โดยมีรู้ว่าตนเองคือผู้บริหารอาคาร และปัจจุบันได้กลายเป็นอาชีพบริการที่สำคัญ รูปแบบการบริหารอาคารของไทยมีอยู่ 2 รูปแบบหลักๆ ที่ใช้กันในปัจจุบัน คือ

### 2.4.2 เจ้าของอาคารเป็นผู้บริหารจัดการ

ลักษณะของการบริหารอาคาร โดยเจ้าของบริหารเองนั้น จะสังเกตได้ง่ายกับอาคารขนาดเล็ก เช่น หอพัก อพาร์ทเมนต์ ห้องเช่าต่างๆ คือ เจ้าของจะเป็นผู้ดูแลเองทั้งหมดตั้งแต่ การเก็บค่าใช้จ่าย ซ่อมแซมความเสียหายที่เกิดขึ้น ดูแลรักษาความสะอาด โดยอาจจ้างพนักงานไว้เอง หรืออาจจ้างเป็นครั้งคราวก็ได้

### 2.4.3 เจ้าของอาคารจ้างนิติบุคคลภายนอกหรือบุคคลภายนอกเป็นผู้บริหารจัดการ

ลักษณะการบริหารอาคารนี้จะหมายถึง นิติบุคคลหรือบุคคลที่เจ้าของอาคารได้ว่าจ้างและแต่งตั้งให้เป็นผู้บริหารอาคารจัดการอาคาร มีหน้าที่ดำเนินการ ควบคุมและจัดการทรัพย์สิน การบริการและการบำรุงรักษาในทุกเรื่องที่เกี่ยวข้องกับอาคาร หรือตามแต่จะตกลงกัน โดยผู้บริหารจัดการอาคารจะมีทีมงานในการดำเนินการ และจะต้องจัดทำรายงานให้เจ้าของอาคารตลอดระยะเวลาที่บริหาร รวมทั้งรับนโยบายการบริหารจากเจ้าของอาคารอีกด้วย การใช้ลักษณะการบริหารอาคารแบบนี้จะพบเห็นได้จากอาคารขนาดใหญ่ทั่วไป ทั้งที่เป็นอาคารพักอาศัย และอาคารพาณิชย์

หลักการบริหารอาคารหลัก ในการบริหารอาคารจะเหมือนกับการบริหารงาน โดยทั่วไป กล่าวคือ ผู้บริหารอาคารจะต้องดำเนินการจัดการให้มีค่าใช้จ่ายในการซ่อมบำรุงอาคารน้อยที่สุด ดำเนินการวางระบบการบริหารอาคาร เพื่อให้เกิดการจัดการที่ง่ายและรวดเร็วเมื่อเกิดปัญหา บริหารจัดการในทุกเรื่องที่เกี่ยวข้องกับอาคาร เพื่อให้ผู้อยู่อาศัยในอาคารมีความพึงพอใจสูงสุด บริหารและจัดการในเรื่องของรายได้ให้เข้ามามากที่สุดและให้มีรายจ่ายน้อยที่สุด

บทบาทของผู้บริหารจัดการอาคาร โดยทั่วไปแล้วผู้บริหารจัดการอาคารมักถูกมองข้ามความสำคัญไป อาจเนื่องจากผู้เกี่ยวข้องในงานก่อสร้างมีมุมมองว่า ความสำคัญของอาคารอยู่ในช่วงออกแบบและทำการก่อสร้าง ซึ่งอาคารได้ผ่านช่วงก่อสร้างจนถึงช่วงที่อาคารเปิดใช้งานแล้ว ดังนั้นช่วงการเปิดใช้อาคารนี้หากมีการควบคุมดูแลไม่ดีหรือบุคลากรไม่มีความสามารถที่จะใช้สิ่งอำนวยความสะดวกของอาคารได้อย่างมีประสิทธิภาพ เท่ากับว่าสิ่งอำนวยความสะดวกที่เจ้าของอาคารได้ลงทุนไปนั้น ใช้งานไม่มีประสิทธิภาพและไม่คุ้มค่าแก่การลงทุน เจ้าของอาคารสูงที่เห็นความสำคัญส่วนนี้มักจะจัดจ้างผู้บริหารจัดการอาคาร ให้เข้ามามีบทบาทก่อนงานก่อสร้างแล้วเสร็จ เพื่อที่จะให้ผู้บริหารจัดการอาคาร ได้เข้าร่วมตรวจสอบรับมอบงาน อบรมวิธีการใช้งานจากผู้รับเหมาและผู้บริหารงานก่อสร้าง ซึ่งจะมีข้อดีกว่าให้ผู้บริหารจัดการอาคารเข้ามามีบทบาทหลังจากงานก่อสร้างแล้วเสร็จ

## 2.5 ลักษณะการจัดการงานบำรุงรักษา

ภาพรวมของการจัดการงานบำรุงรักษาจะเห็นข้อแตกต่างระหว่างการจัดการงานบำรุงรักษาที่ดีและไม่ดีได้ง่าย จึงได้สรุปลักษณะสำคัญของการจัดการงานบำรุงรักษาที่ดีและไม่ดีดังนี้

### 2.5.1. ลักษณะของการจัดการบำรุงรักษาที่ไม่ดี

- การบำรุงรักษาอุปกรณ์ทั้งหมดส่วนใหญ่จะขึ้นอยู่กับช่างที่ชำนาญงานเฉพาะเป็นหลัก
- ไม่มีการเก็บข้อมูลของเครื่องจักรอุปกรณ์และประวัติของเครื่องจักรอุปกรณ์ส่วนใหญ่จะอาศัยการจดจำของช่างซ่อมบำรุง เมื่อมีช่างลาออกไปประวัติส่วนที่ช่างผู้นั้นจดจำก็จะหายไปด้วย
- ไม่สามารถที่จะประเมินและควบคุมค่าใช้จ่ายในการบำรุงรักษาอุปกรณ์ได้
- มีการทำงานล่วงเวลาของช่างซ่อมบำรุง
- ผู้บริหารมักมองและคิดว่าการบำรุงรักษาเป็นค่าใช้จ่ายและเป็นสิ่งที่เลวร้ายที่จำใจต้องดำเนินการ
- เวลาของช่างซ่อมบำรุงส่วนใหญ่ถูกใช้ไปสำหรับงานบำรุงรักษาและอุปกรณ์ที่ไม่ได้วางแผนไว้ล่วงหน้า
- พนักงานควบคุมอุปกรณ์มีหน้าที่ดูแลอุปกรณ์ โดยรอว่าหากผิดปกติจะแจ้งหน่วยบำรุงรักษาเพียงอย่างเดียวโดยไม่คำนึงถึงสภาพของอุปกรณ์

### 2.5.2 ลักษณะของการบำรุงรักษาที่ดี

- การบำรุงรักษาอุปกรณ์เป็นกิจกรรมที่ผู้บริหารยอมรับว่าเป็นส่วนหนึ่งและเป็นส่วนที่จำเป็นต้องมี

- แผนกบำรุงรักษามุ่งเน้นไปที่การดำเนินงานเพื่อให้ได้ความพร้อมใช้งานของอุปกรณ์ ซึ่งเป็นการเพิ่มความเชื่อถือของอุปกรณ์ให้สูงขึ้น

- มีการให้ความสำคัญต่อการวิเคราะห์หาสาเหตุของการหยุดทำงานของอุปกรณ์

- มีความมุ่งมั่นที่จะดำเนินการตามกิจกรรมของการบำรุงรักษาอุปกรณ์ที่ได้

วางแผนไว้

- มีการให้ความสำคัญและมุ่งเน้นในการฝึกอบรมพนักงานที่เกี่ยวข้องทั้งหมดในเรื่องที่จำเป็นต่อการปฏิบัติงานบำรุงรักษาและการใช้อุปกรณ์

- มีกระบวนการปรับปรุงการดำเนินการบำรุงรักษาอย่างต่อเนื่อง

- พนักงานควบคุมอุปกรณ์ได้เข้าไปมีส่วนร่วมในการบำรุงรักษาอุปกรณ์นั้นๆ

### 2.5.3 ปัญหาของระบบการจัดการงานบำรุงรักษาแบบใช้กระดาษ

ระบบการจัดการบำรุงรักษาแบบใช้กระดาษหรือระบบที่ใช้พนักงานเป็นผู้เก็บรวบรวมประมวลผล และวิเคราะห์ข้อมูล แล้วนำไปใช้ในการวางแผนการปฏิบัติการควบคุมและการประเมินผลการดำเนินงานบำรุงรักษาหรือที่เรียกว่าระบบ Manual โดยทั่วไปจะประกอบด้วยระบบย่อยๆ คือ ระบบทะเบียนและประวัติอุปกรณ์จะเป็นระบบการจัดทำทะเบียนซึ่งประกอบด้วยรหัสและรายละเอียดที่สำคัญของอุปกรณ์ และการเก็บประวัติการซ่อมและการบำรุงรักษาอุปกรณ์นั้นๆ การติดตามประวัติการโยกย้ายของอุปกรณ์ ระบบการวางแผนงานบำรุงรักษา ซึ่งเป็นการกำหนดกิจกรรมของการบำรุงรักษาป้องกันที่จะดำเนินการต่อแต่ละชิ้นตามช่วงระยะเวลาต่างๆ และระบบใบสั่งงานซึ่งเป็นการมอบหมายงานซ่อมและบำรุงรักษาให้พนักงานไปดำเนินงานและเป็นการรายงานการซ่อมแซมและบำรุงรักษาที่ได้ไปดำเนินการแล้วเพื่อนำมาวิเคราะห์และเก็บเป็นประวัติต่อไปโดยปัญหาของระบบการจัดการงานบำรุงรักษาข้างต้นที่สำคัญคือ

- การฝึกอบรมให้พนักงานมีวินัยในการรักษาให้ข้อมูลต่างๆ ที่เกี่ยวข้องทันสมัยอยู่เสมอ และผู้บริหารสามารถเรียกดูข้อมูลต่างๆ เพื่อประกอบการตัดสินใจได้ทันที มักเป็นเรื่องที่กระทำยากก่อนข้างยาก

- หัวหน้างานมักต้องใช้ความพยายามค่อนข้างมากในการจัดทำเอกสารเพื่อมอบหมายงานบำรุงรักษาที่ครบถ้วนให้แก่บุคลากรที่เกี่ยวข้องรวมทั้งการควบคุมและวิเคราะห์ผลของการทำงาน

- ช่วงซ่อมบำรุงส่วนใหญ่ไม่ค่อยเต็มใจและมักจะหลีกเลี่ยงการทำงานที่เกี่ยวกับเอกสาร (เช่น การบันทึกข้อมูลที่กำหนดลงในแบบพิมพ์และการเขียนรายงาน เป็นต้น)

- ผู้ที่เกี่ยวข้องมักต้องใช้ความพยายามค่อนข้างมากในการจัดเก็บรวบรวมและประเมินผลข้อมูล เพื่อให้ได้สถิติและตัวชี้วัดที่มีความหมายซึ่งสามารถแสดงถึงสถานะและปัญหาที่เกิดขึ้นของระบบการจัดการงานบำรุงรักษาที่ทันต่อเหตุการณ์ ดังนั้นจึงมักพบว่าระบบการจัดการงานบำรุงรักษาแบบใช้กระดาษที่เป็นอยู่ส่วนใหญ่จะเสียเวลาในการดำเนินการมาก ซึ่งหากไม่ได้มีการเอาใจใส่อย่างจริงจังจากผู้บริหารทุกระดับระบบการจัดการดังกล่าวก็มักจะไม่สามารถปรับปรุงและพัฒนา และถูกปล่อยปละละเลยจนไม่สามารถนำเอาข้อมูลที่มีอยู่มาใช้ประโยชน์ได้เลย[6]

## 2.6 ระบบการจัดการงานบำรุงรักษาอุปกรณ์โดยใช้โปรแกรมคอมพิวเตอร์

(Computerized Maintenance Management System)

ระบบการจัดการงานบำรุงรักษาอุปกรณ์ โดยใช้โปรแกรมคอมพิวเตอร์ [7] ที่นิยมเรียกกันสั้นๆ โดยใช้ตัวย่อภาษาอังกฤษว่าระบบ MMS หรือถ้าเรียกให้ถูกต้องควรจะเป็น CMMS ที่มาจากชื่อเต็มคือ Computerized Maintenance Management System ในปัจจุบันได้มีการนำมาใช้ในหน่วยงานบำรุงรักษาหลายๆ หน่วยงาน ประกอบกับเมื่อเครื่องคอมพิวเตอร์ส่วนบุคคลมีราคาถูกลงและมีความสามารถสูงขึ้น ทำให้ผู้บริหารหันมาสนใจที่จะนำเอาระบบ CMMS มาใช้ แต่ผู้บริหารส่วนใหญ่มักคิดว่าระบบ CMMS จะเป็นคำตอบของปัญหาที่มีอยู่ทั้งหมดในด้านการจัดการงานบำรุงรักษา ซึ่งยังเป็นความเข้าใจที่คลาดเคลื่อน จึงทำให้หน่วยงานที่นำเอาระบบ CMMS มาใช้ส่วนใหญ่ไม่ประสบความสำเร็จ และต้องเลิกใช้ไปในที่สุดหรือสามารถใช้ประโยชน์จากระบบที่นำมาใช้ได้เพียงเล็กน้อยเท่านั้น

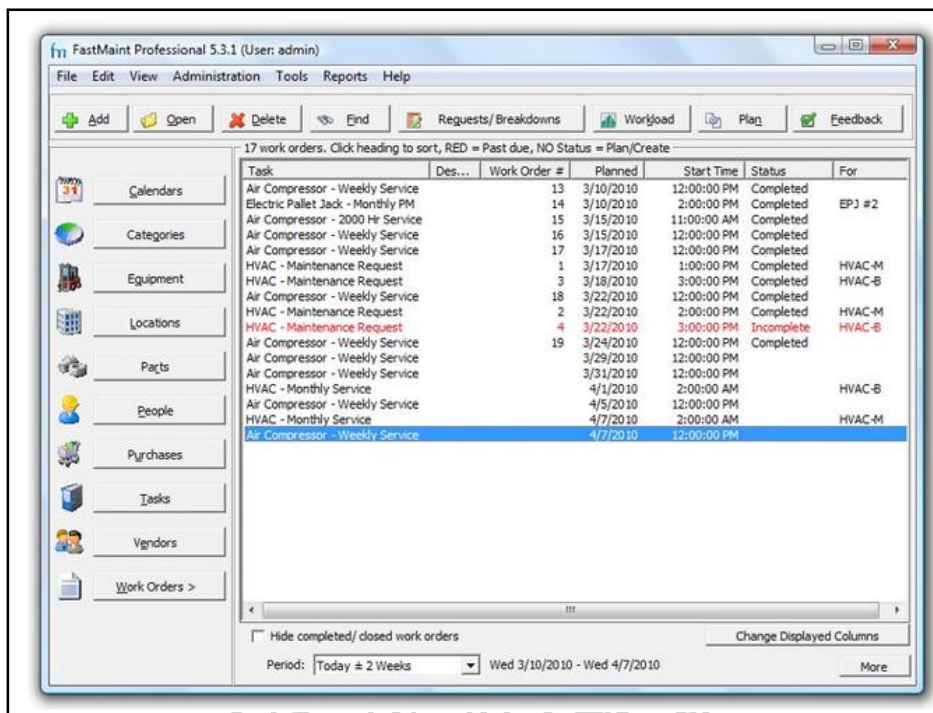
การศึกษาและทำความเข้าใจกับรายละเอียดต่างๆ ของระบบ CMMS ได้แก่ ความสามารถของระบบ รายละเอียดของงานแต่ละงานที่ระบบสามารถทำได้และเงื่อนไขในการทำงานของระบบ รวมทั้งสถานะของงานบำรุงรักษาอุปกรณ์ที่ใช้พนักงานเป็นผู้เก็บประมวลผลและวิเคราะห์ข้อมูล หรือที่เรียกว่าระบบใช้กระดาษ ที่ดำเนินการอยู่ นับว่าเป็นสิ่งสำคัญสูงสุดที่ผู้บริหารต้องนำมาวิเคราะห์ก่อนว่าโปรแกรมของระบบ CMMS เมื่อนำมาใช้งาน จะกำหนดขั้นตอนของการนำเอา

ระบบมาใช้ รูปแบบและขั้นตอนของการทำงานเดิมที่จำเป็นต้องปรับปรุงเปลี่ยนแปลงและเพิ่มเติมอย่างไร รวมทั้งประโยชน์ที่คาดว่าจะได้รับจากโปรแกรมมาน้อยเพียงไร เพื่อหลีกเลี่ยงความล้มเหลวที่จะเกิดขึ้น ปัญหาทั่วไปของระบบการจัดการงานบำรุงรักษาที่ใช้กระดาษหรือดำเนินการโดยพนักงาน ความสามารถและประโยชน์ที่คาดว่าจะได้รับจากระบบ CMMS และรูปแบบการทำงานโดยระบบ CMMS เพื่อช่วยให้ผู้บริหารสามารถทำความรู้จักกับระบบ CMMS ได้ง่ายและรวดเร็วยิ่งขึ้น

#### 2.6.1 ความสามารถของ CMMS

CMMS คือ ระบบสารสนเทศที่พัฒนาขึ้นสำหรับงานด้านการบำรุงรักษาประกอบด้วย การจัดเก็บข้อมูล การประมวลผลข้อมูล และการรายงานเพื่อสนับสนุนในการวางแผน ปฏิบัติติดตาม การวิเคราะห์ข้อมูลและประเมินผลงานด้านสมรรถนะการบำรุงรักษา โปรแกรมของระบบ CMMS บนจอคอมพิวเตอร์ ดังรูปที่ 3 ในระยะแรกได้ออกแบบมาใช้เป็นเครื่องมือเพื่อช่วยในการเก็บรวบรวม ประมวลผลข้อมูล และการจัดทำรายงาน รวมทั้งการค้นหาข้อมูลเดิมที่เกี่ยวกับระบบการจัดการงานบำรุงรักษา สามารถทำได้ง่ายและสะดวกรวดเร็ว ซึ่งก็จะแยกเป็นระบบย่อยๆ เช่นเดียวกับระบบที่ดำเนินการโดยพนักงานหรือระบบที่ใช้กระดาษและต่อมาก็ได้มีการผนวกเอา ระบบการสนับสนุนการบำรุงรักษาต่างๆ เข้าไว้ในระบบ CMMS ด้วย ได้แก่ ระบบการจัดการอะไหล่ที่รวมทั้งระบบการจัดเก็บ การเบิกจ่าย และการจัดซื้ออะไหล่ เข้าไว้ด้วยกัน ระบบการเก็บข้อมูลอุปกรณ์เพื่อใช้ในการหาประสิทธิภาพโดยรวมของอุปกรณ์ (Overall Equipment Effectiveness) ระบบการวิเคราะห์สภาพอุปกรณ์และระบบอื่นๆ ที่เกี่ยวข้องอีก ทำให้โปรแกรมของระบบ CMMS มักจะไม่ออกแบบโปรแกรมที่มีรูปแบบเดียวตายตัว แต่จะออกแบบโปรแกรมเป็น โปรแกรมชุดย่อยๆ (Module) ที่สามารถเสนอให้ผู้เลือกใช้ไปใช้ได้ตามความต้องการและตามความเหมาะสม





ภาพที่ 3 แสดงโปรแกรมของระบบ CMMS บนจอคอมพิวเตอร์ [8]

## 2.6.2 ข้อพิจารณาก่อนการนำระบบ CMMS มาใช้

การนำเอาระบบ CMMS มาใช้นั้นจะต้องทำความเข้าใจในเบื้องต้นเสียก่อนว่าระบบ CMMS เป็นเพียงเครื่องมือที่จะช่วยให้การจัดการงานบำรุงรักษาอุปกรณ์สามารถทำได้สะดวกและรวดเร็วยิ่งขึ้นเท่านั้น การจัดเก็บข้อมูลที่จำเป็นซึ่งทำอยู่เดิมก็ยังคงต้องทำต่อไป และถ้ามีความจำเป็นที่จะต้องจัดเก็บข้อมูลเพิ่มเติมที่ต้องการสำหรับระบบ CMMS ที่จะนำมาใช้ในการประมวลผลตามที่ได้กำหนดไว้ก็จะต้องดำเนินการด้วย และจะต้องมีการป้อนข้อมูลเข้าเครื่องคอมพิวเตอร์ที่จะต้องทำเป็นประจำอีกด้วย ส่วนการประมวลผล การวิเคราะห์ข้อมูลและการจัดทำรายงาน รวมทั้งการเรียกดูข้อมูลและสถิติต่างๆ แม้สามารถทำได้ง่ายเมื่อใช้ระบบ CMMS แต่ผลลัพธ์ก็ยังคงขึ้นอยู่กับความถูกต้อง ครบถ้วน และทันกับเวลา ผลลัพธ์ก็จะน่าเชื่อถือ แต่ในทางตรงข้ามกันถ้าข้อมูลไม่ถูกต้อง หรือไม่ครบถ้วน หรือไม่ทันกับเวลา ผลลัพธ์ที่ได้ก็จะไม่น่าเชื่อถือไปด้วย



ลักษณะของการจัดการงานบำรุงรักษาอุปกรณ์ทั้งที่ดีและไม่ดีนั้นก็ช่วยให้ผู้บริหารได้ประเมินสถานะของการจัดการงานบำรุงรักษาอุปกรณ์เสียก่อน เพื่อให้ตระหนักถึงความจำเป็นในการปรับปรุง เปลี่ยนแปลงและเพิ่มเติมระบบการจัดการงานบำรุงรักษาที่เป็นอยู่เดิม ทั้งนี้เนื่องจากถ้าระบบการจัดการงานบำรุงรักษาแบบใช้กระดาษเดิมคืออยู่แล้วก็มักจะไม่ต้องมีการปรับปรุง เปลี่ยนแปลง และเพิ่มเติมในส่วนที่จำเป็นของระบบที่ใช้อยู่เดิมมากนัก ก็สามารถที่จะรองรับการนำเอาระบบ CMMS มาใช้ได้ เป็นผลให้การนำเอาระบบ CMMS มาใช้มักจะประสบความสำเร็จได้ไม่ยากนักในทางตรงข้ามกัน ถ้าระบบการจัดการงานบำรุงรักษาแบบใช้กระดาษที่ใช้อยู่เดิมเป็นระบบการจัดการที่ไม่ดี และการนำเอาระบบ CMMS มาใช้นั้นก็ต้องมีความมุ่งมั่นและความตั้งใจที่จะปรับปรุง เปลี่ยนแปลง และเพิ่มเติมในส่วนที่จำเป็นของระบบที่ใช้อยู่เดิมให้สามารถรองรับและสอดคล้องกับระบบ CMMS ที่จะนำมาใช้ จึงส่งผลให้การนำเอาระบบ CMMS มาใช้ประสบความสำเร็จ แต่หากไม่ดำเนินการปรับปรุง เปลี่ยนแปลง และเพิ่มเติมในส่วนที่จำเป็นของระบบที่ใช้อยู่เดิมเลย แล้วหวังว่าระบบ CMMS ที่จะนำมาใช้ จะสามารถแก้ปัญหาและปรับปรุงระบบการจัดการงานบำรุงรักษาที่เป็นอยู่เดิมให้ดีขึ้นโดยอัตโนมัติแล้ว การนำเอาระบบ CMMS มาใช้ก็จะไม่มีทางที่จะประสบความสำเร็จได้อย่างแน่นอน

## 2.7 การสำรวจลักษณะขั้นตอนการบริหารจัดการอาคารสถานที่ กรณีศึกษา

ได้ทำการสำรวจกระบวนการบริหารจัดการบำรุงรักษาอาคาร จากหน่วยงานจำนวน 3 แห่ง และสรุปได้ว่าในหน่วยงานแต่ละที่นั้น มีความแตกต่างกันในเรื่องของโครงสร้างฝ่ายงาน แต่ในส่วนของขั้นตอนกระบวนการบำรุงรักษาอาคารนั้นมีผู้เกี่ยวข้องและขั้นตอนกระบวนการหลักๆ ดังนี้

- 1.) ผู้เกี่ยวข้องกับการบำรุงรักษาอาคารประกอบไปด้วย
  - 1.1 ผู้แจ้งซ่อม
  - 1.2 ผู้รับแจ้ง
  - 1.3 หัวหน้างานฝ่ายซ่อมบำรุง
  - 1.4 ช่างซ่อมบำรุงในงานเฉพาะด้านต่างๆ

2.) ขั้นตอนบริหารจัดการบำรุงรักษาอาคาร โดยสรุปจากการสำรวจหน่วยงานต่างๆ

2.1 การแจ้งซ่อมผ่านทางโทรศัพท์ หรือ แบบฟอร์ม

2.2 หัวหน้าฝ่ายงานซ่อมบำรุงจ่ายงานให้กับช่าง

2.3 ช่างคู่มืองานเพื่อประเมินความเสียหาย

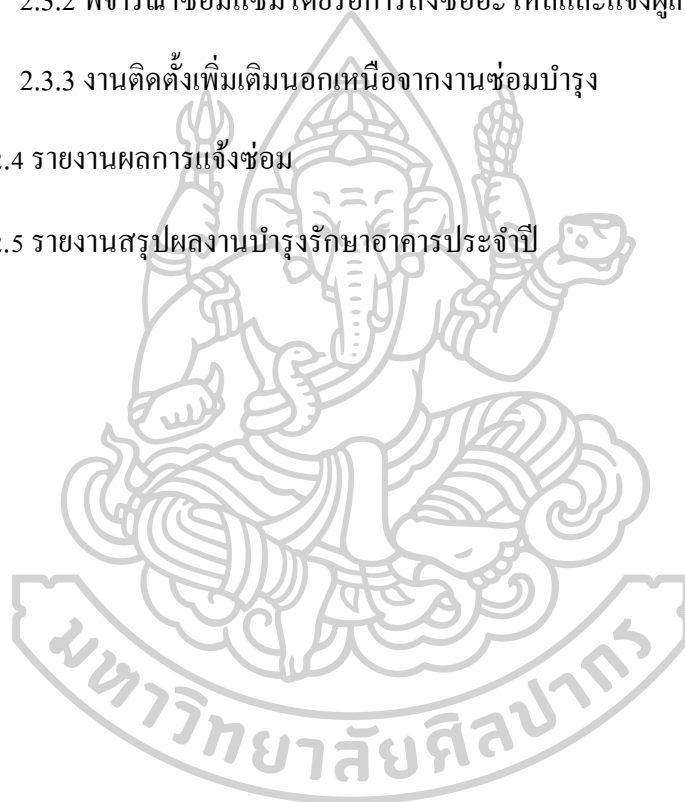
2.3.1 พิจารณาซ่อมแซมได้ทันทีใช้อะไหล่ที่มีอยู่แล้ว

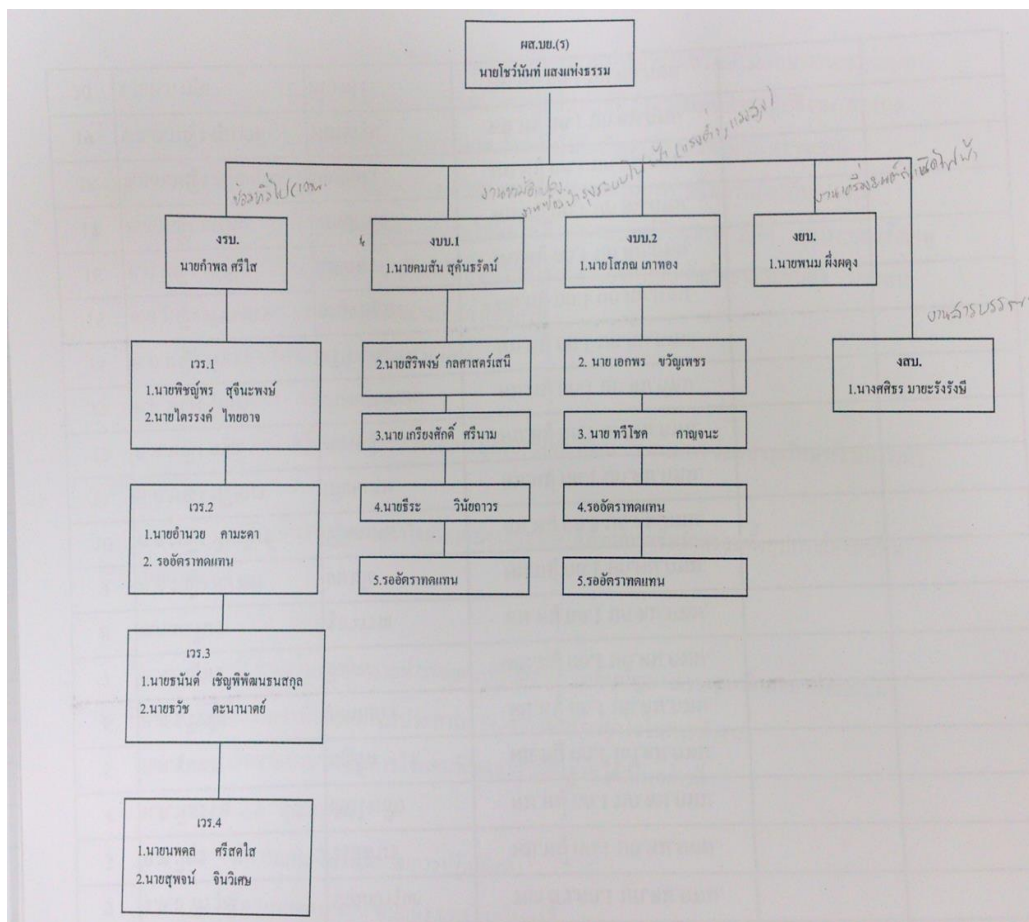
2.3.2 พิจารณาซ่อมแซมโดยรอการสั่งซื้ออะไหล่และแจ้งผู้เกี่ยวข้อง

2.3.3 งานติดตั้งเพิ่มเติมนอกเหนือจากงานซ่อมบำรุง

2.4 รายงานผลการแจ้งซ่อม

2.5 รายงานสรุปผลงานบำรุงรักษาอาคารประจำปี

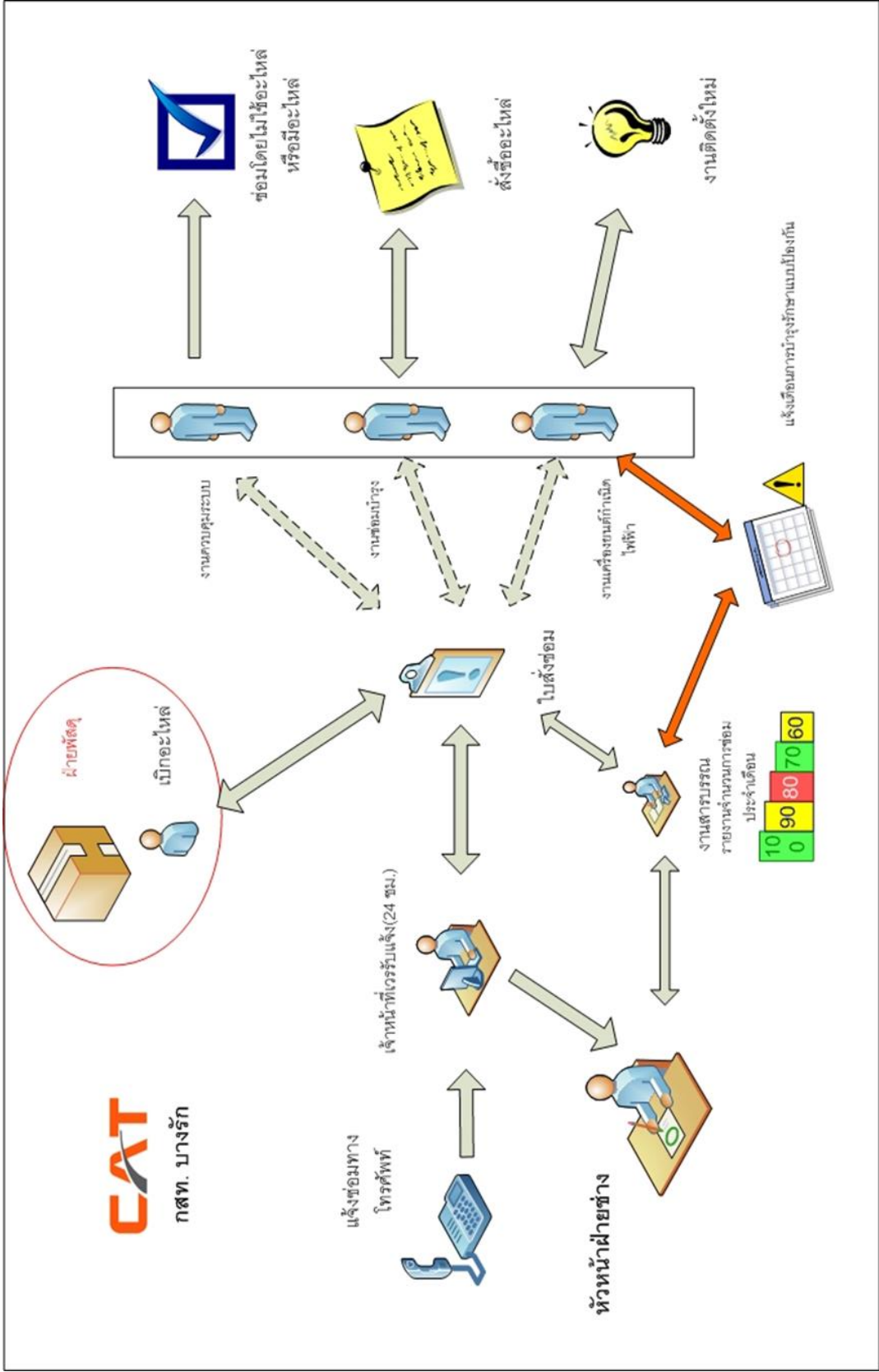




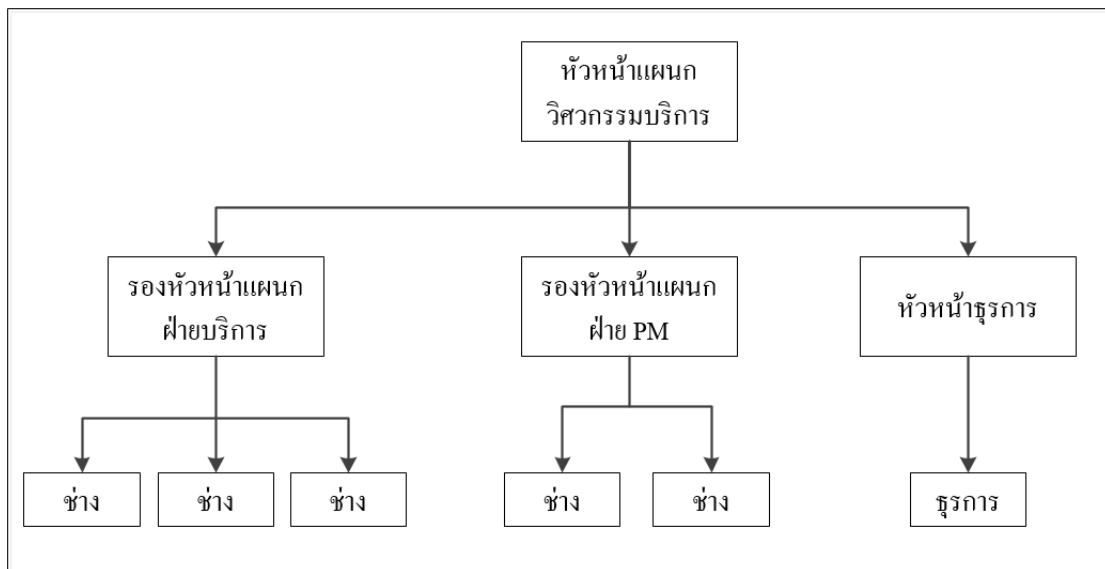
ภาพที่ 4 แสดงความสัมพันธ์ของสายงาน บริษัท กสท. โทรคมนาคมจำกัด (มหาชน) เขตบางรัก จากภาพแสดงความสัมพันธ์ของสายงาน บริษัท กสท. โทรคมนาคมจำกัด (มหาชน) เขตบางรัก อธิบายโครงสร้างได้ดังนี้

### 1 ผู้อำนวยการสำนักงาน

- หัวหน้างานบริการซ่อมบำรุงทั่วไป (จัดเวร )
- หัวหน้างานซ่อมบำรุงระบบไฟฟ้าแรงดันต่ำ และ แรงดันสูง
- หัวหน้างานเครื่องยนต์และเครื่องกำเนิดไฟฟ้า
- หัวหน้างานสารบรรณ



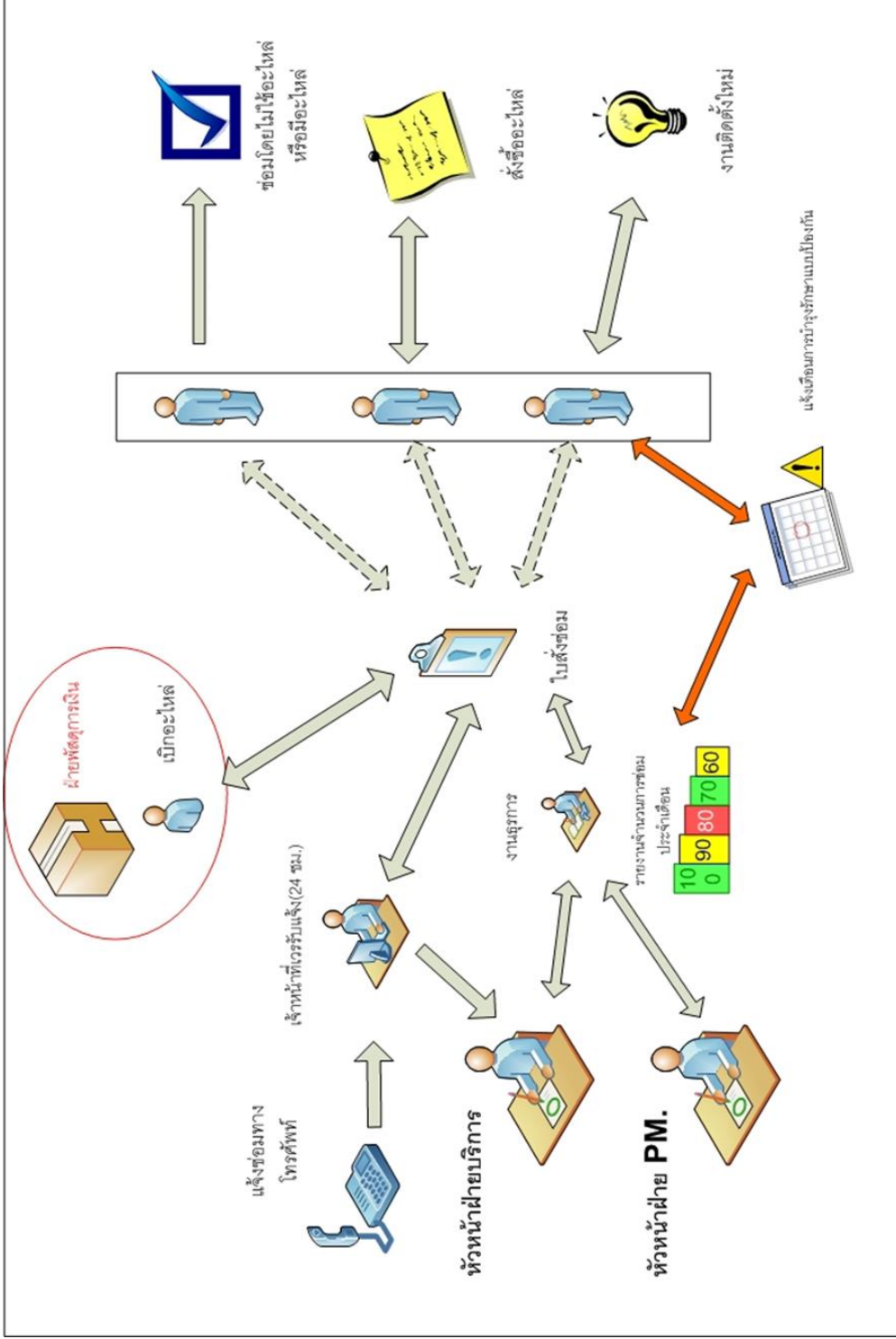
ภาพที่ 5 แสดงขั้นตอนการดำเนินงานซ่อมบำรุง บริษัท กสท. โทรคมนาคมจำกัด (มหาชน) เขตบางรัก



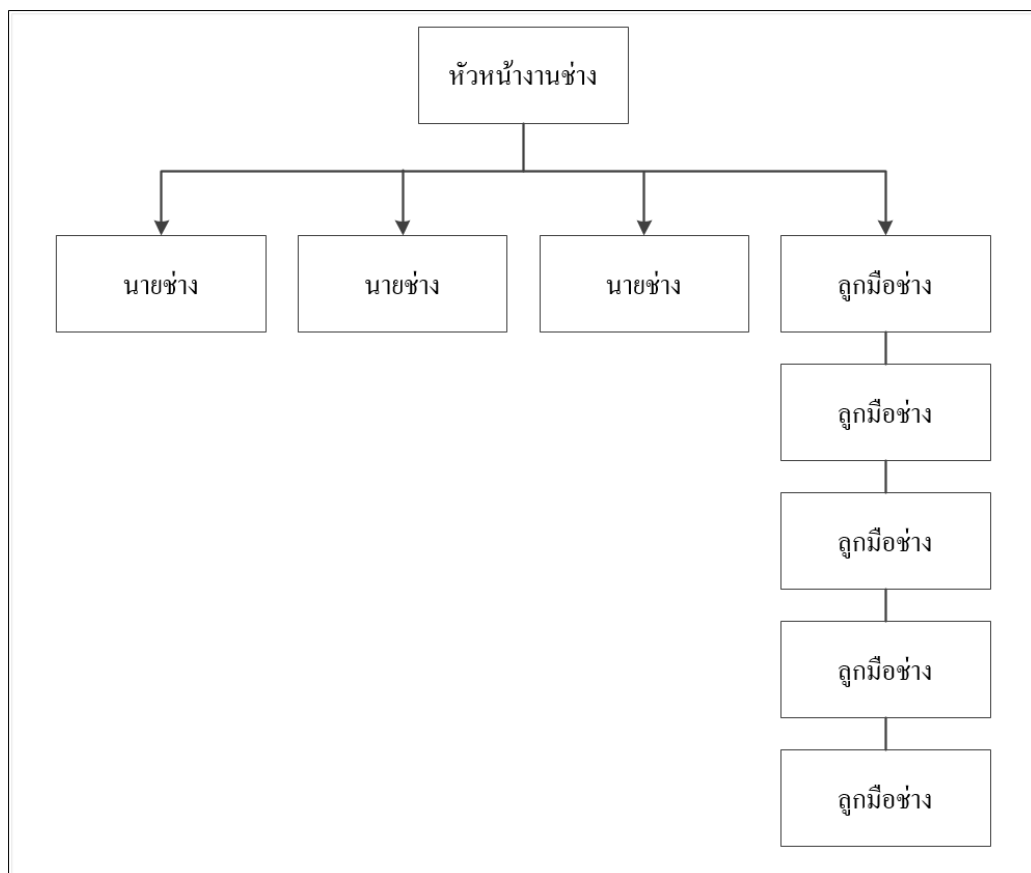
ภาพที่ 6 แผนผังการบังคับบัญชาแผนกวิศวกรรมบริการ รพ.นครพนม  
จากภาพแสดงความสัมพันธ์ของสายงานการบังคับบัญชาแผนกวิศวกรรมบริการ รพ.นครพนม  
ซึ่งอธิบายโครงสร้างได้ดังนี้

#### 1 ผู้อำนวยการสำนักงาน

- หัวหน้าแผนกวิศวกรรมบริการ
- รองหัวหน้าแผนกฝ่ายบริการ
- รองหัวหน้าแผนกฝ่าย PM.
- หัวหน้าธุรการ
- ช่างซ่อมบำรุง



ภาพที่ 7 ขั้นตอนการทำงานแผนกวิศวกรรมบริการ รพ.นครชน

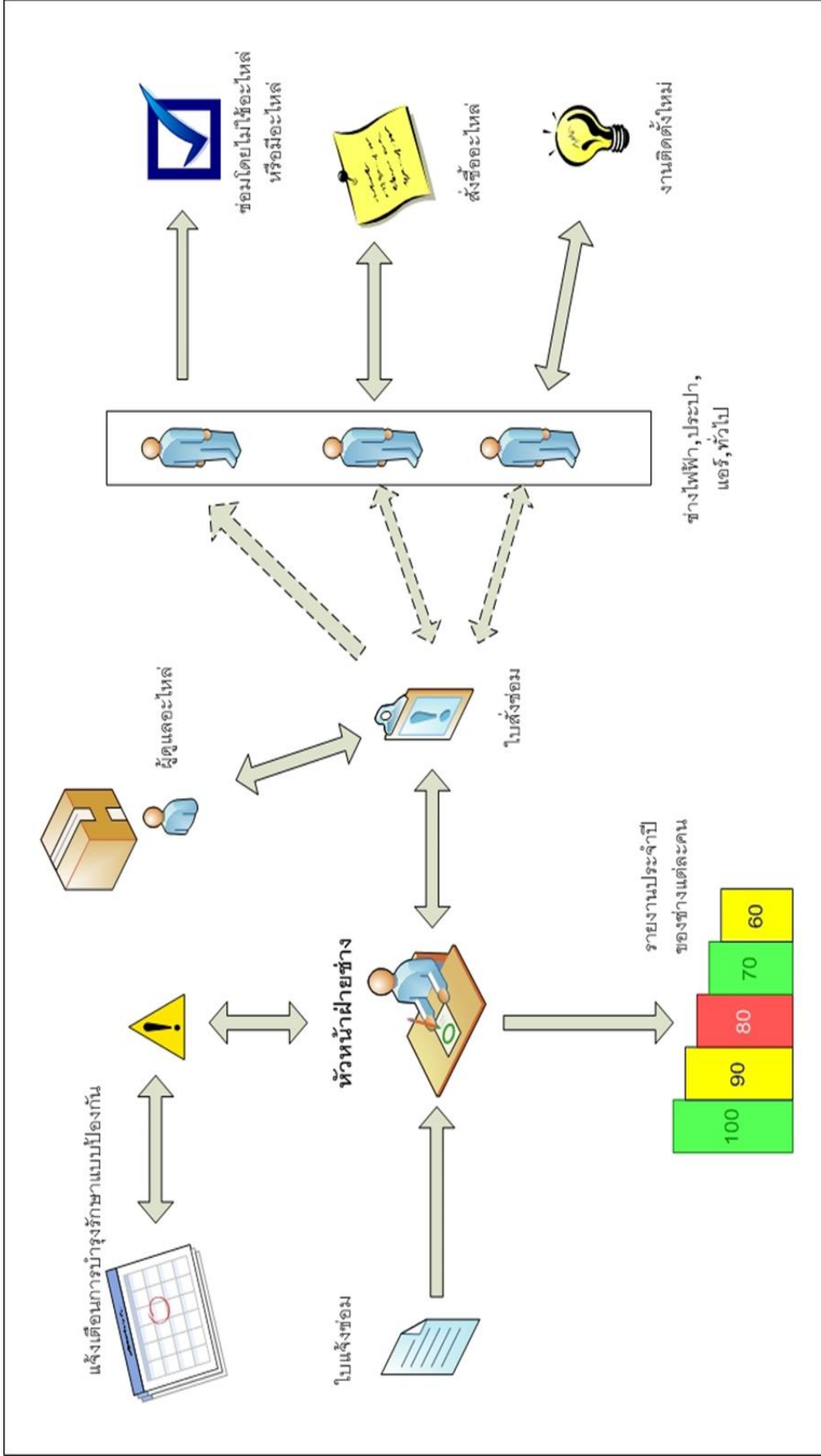


ภาพที่ 8 ความสัมพันธ์ของฝ่ายงานช่าง มหาวิทยาลัยศิลปากร วิทยาเขตสารสนเทศเพชรบุรี จากภาพแสดงความสัมพันธ์ของสายงานการบังคับบัญชาของสายงานฝ่ายงานช่าง มหาวิทยาลัยศิลปากร วิทยาเขตสารสนเทศเพชรบุรี ซึ่งอธิบายโครงสร้างได้ดังนี้

1 หัวหน้างานช่าง

- ตำแหน่งนายช่างจำนวน 3 ตำแหน่ง
- ลูกมือช่างจำนวน 5 ตำแหน่ง





ภาพที่ 9 ขั้นตอนการทำงานของฝ่ายช่าง มหาวิทยาลัยศิลปากร วิทยาเขตสารสนเทศเพชรบุรี

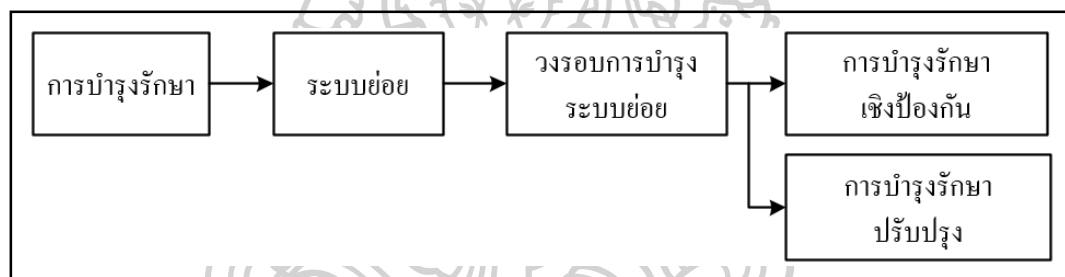


## 2.8 แนวคิดและทฤษฎีฐานข้อมูล (Database)

ฐานข้อมูล [6] หมายถึง การเก็บรวบรวมกลุ่มของข้อมูลไว้ด้วยกัน เพื่อให้บุคคลและงานประยุกต์ต่างๆ สามารถดำเนินการอย่างใดอย่างหนึ่ง เช่น การเพิ่มข้อมูล ลบข้อมูล แก้ไขข้อมูล และการดึงข้อมูลมาใช้งาน เป็นต้น ฐานข้อมูลแบ่งตามลักษณะการใช้งานได้ 3 แบบ คือ

### 2.8.1 ฐานข้อมูลแบบแตกสาขา (Hierarchical Database)

เป็นฐานข้อมูลรุ่นแรก ปัจจุบันไม่นิยมใช้เนื่องจากมีความยุ่งยากในการสร้างและใช้งาน โดยหลักการของฐานข้อมูลจะเริ่มจากสมาชิกระดับต้น ซึ่งไปยังข้อมูลในระดับล่าง โดยการชี้ข้อมูลจะต้องทำเป็นลำดับตามผู้ออกแบบกำหนด ข้อเสียของฐานข้อมูลชนิดนี้คือ ใช้งานยากมีข้อจำกัดว่าสมาชิกระดับล่าง จะเกี่ยวข้องกับสมาชิกระดับบน ได้เพียง 1 ความสัมพันธ์เท่านั้น ดังรูปที่ 10

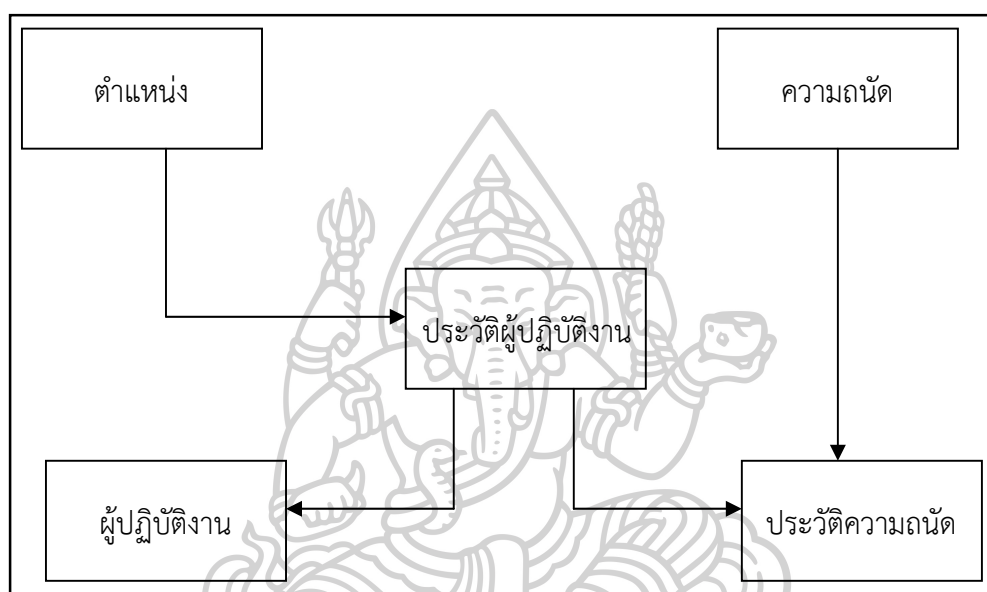


ภาพที่ 10 ตัวอย่างฐานข้อมูลแบบแตกสาขา

จากภาพ การที่จะทราบรายละเอียดการปฏิบัติงาน ที่เกี่ยวข้อง ใต้นั้น จะต้องผ่านระบบย่อย วงรอบการซ่อมบำรุงระบบย่อย และจะมีความสัมพันธ์กัน ได้เพียงหนึ่งความสัมพันธ์เท่านั้น

### 2.8.2 ฐานข้อมูล แบบเครือข่าย (Network Database)

ข้อเสียของฐานข้อมูลแบบแตกสาขา ถูกแก้ด้วยระบบเครือข่าย คือ สมาชิกลูก (Child) สามารถเกี่ยวข้องกับสมาชิกพ่อแม่ (Parent) ได้มากกว่า 1 แต่ก็ยังกำหนดความสัมพันธ์ตั้งแต่เริ่มสร้างข้อมูล และไม่สามารถเปลี่ยนแปลงในระหว่างการทำงานได้ ซึ่งแสดงได้ดังรูปที่ 11

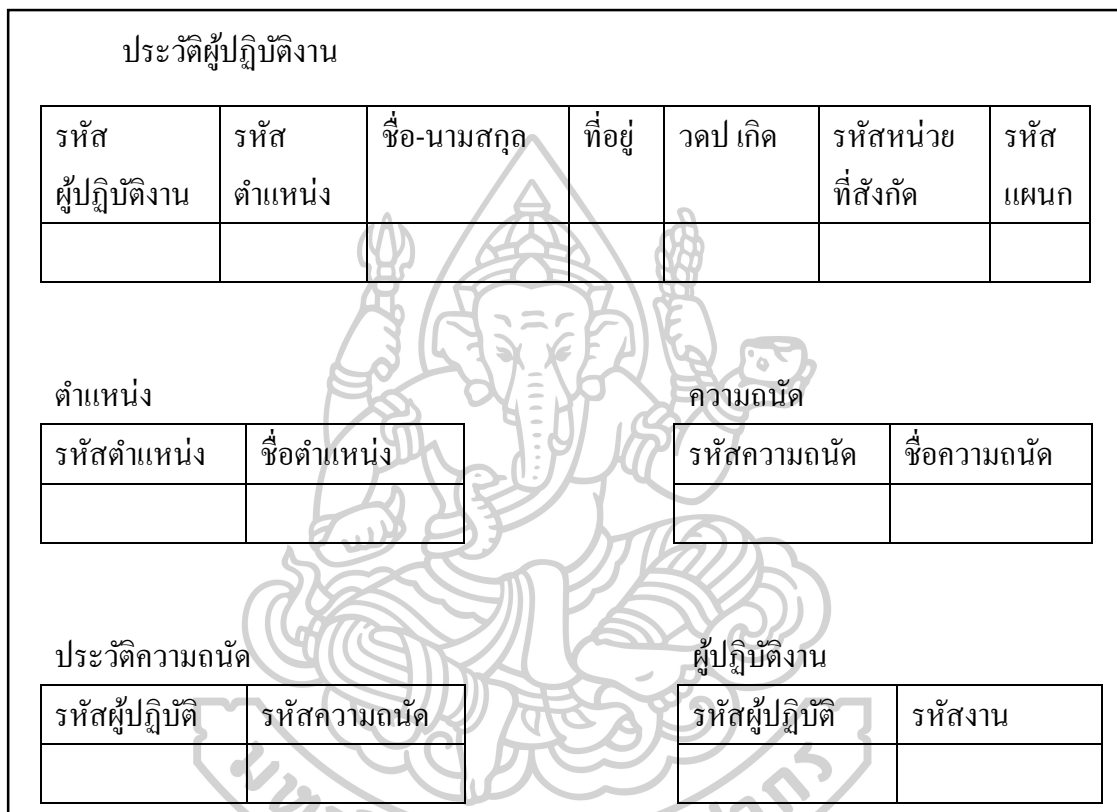


ภาพที่ 11 ฐานข้อมูลแบบเครือข่าย

จากรูปจะเห็นว่า ประวัติความถนัดซึ่งเป็นสมาชิกลูก สามารถมีความสัมพันธ์กับสมาชิกแม่ได้มากกว่าหนึ่งความสัมพันธ์ ได้แก่ ประวัติผู้ปฏิบัติงาน และรายละเอียดความถนัด โดยที่ผู้ปฏิบัติงานหนึ่งคนสามารถมีความถนัด ได้มากกว่าหนึ่ง และยังสามารถปฏิบัติงานได้มากกว่าหนึ่งงานอีกด้วย การค้นหาข้อมูลจากฐานข้อมูลอาจวิ่งตามหัวลูกศรหรือทวนหัวลูกศรก็ได้ เช่น เมื่อต้องการทราบความถนัดของผู้ถัดของผู้ปฏิบัติงาน ก็ทำได้โดยการระบุรหัสและวิ่งตามลูกศรไปที่ประวัติ ความถนัด หรือต้องการชื่อผู้ปฏิบัติงาน อาจวิ่งทวนหัวลูกศรจากตาราง (Table) ประวัติความถนัดไปยังตาราง (Table) ประวัติผู้ปฏิบัติงาน โดยต้องทราบรหัสผู้ปฏิบัติงานก่อน เป็นต้น

### 2.8.3 ระบบฐานข้อมูลแบบสัมพันธ์ (Relation Database)

เป็นฐานข้อมูลที่ได้รับการพัฒนาโดยความสัมพันธ์ของข้อมูลเป็นหลัก โดยจะถูกแทนในลักษณะของตารางและแถวต่างๆ ในตารางจะแสดงค่าของข้อมูลที่มีความสัมพันธ์กัน ชื่อของตารางและชื่อของแถวจะใช้ในการแปลความหมายของค่าต่างๆ ในแต่ละแถวของตารางในฐานข้อมูล ดังรูปที่ 12



ภาพที่ 12 รูปตัวอย่างการเก็บข้อมูลแบบตาราง

จากรูป จะเห็นว่าเราสามารถทราบงานของผู้ปฏิบัติงานได้ โดยการค้นหารหัสผู้ปฏิบัติงานที่ต้องการจากตาราง (Table) ผู้ปฏิบัติงาน ในทางกลับกันถ้าต้องการทราบว่า รหัสงานนี้มีใครเป็นผู้ปฏิบัติบ้างก็เริ่มจากตาราง (Table) ผู้ปฏิบัติงานซึ่งจะทำให้รหัสผู้ปฏิบัติงาน ซึ่งจะทำให้ทราบว่าใครเป็นผู้ปฏิบัติงานนี้บ้าง

### 2.8.4 คุณสมบัติของระบบฐานข้อมูลแบบสัมพันธ์

1. ภาพแบบฐานข้อมูล จะแทนเป็นตาราง (Table) ที่เราสามารถเข้าใจได้ง่าย โดยในตารางจะแบ่งเป็นสดมภ์และเป็นแถว จุดตัดของแถวและสดมภ์จะเรียกว่า เขตข้อมูล (Field) และภายในเขตข้อมูลจะเก็บค่า (Value) ผู้ใช้สามารถเรียกหรือค้นหาข้อมูลต่างๆ ที่ต้องการได้ทุกๆ ค่า และหากเขตข้อมูลใดไม่มีค่าจะกำหนดให้มีค่าว่าง (Null) ไว้

2. ระบบจะต้องมีภาษาระดับสูง เพื่อจัดการข้อมูลที่ถูกเก็บอยู่ในตาราง โดยผู้ใช้งานเพียงระบุว่าต้องการข้อมูลอะไรในฐานข้อมูล โดยไม่ต้องบอกวิธีการประมวลผล ระบบจะจัดการค้นหาเองโดยใช้ภาษาในระดับสูง

3. ภาษาฐานข้อมูล เป็นภาษาที่มีการแบ่งประโยคคำสั่งตามลักษณะงาน ได้ 3 กลุ่ม

ภาษานิยามข้อมูล (Data definition Language : DDL) จะใช้เพื่อกำหนดโครงสร้างข้อมูล กำหนด ตารางข้อมูลเทียบตามทักษะของผู้ใช้ ได้แก่ คำสั่งในการสร้างตาราง เช่น คำสั่ง Create Table เป็นต้น คำสั่งในการสร้างวิว เช่น คำสั่ง Create View เป็นต้น คำสั่งในการสร้างเลขลำดับ เช่น คำสั่ง Create Sequence เป็นต้น คำสั่งในการสร้างดัชนี เช่น คำสั่ง Create Index เป็นต้น คำสั่งในการสร้างคำเหมือน เช่น คำสั่ง Create Synonym เป็นต้น คำสั่งในการแก้ไขโครงสร้างของตาราง เช่น คำสั่ง Alter Table เป็นต้น และคำสั่งในการยกเลิก ตาราง วิว หรือ ดัชนี เช่น คำสั่ง Drop Index/Table/View เป็นต้น

ภาษาจัดการข้อมูล (Data Manipulation Language : TDML) จะใช้เพื่อเปลี่ยนแปลงแก้ไข และการค้นหาข้อมูล ได้แก่ คำสั่งในการเพิ่มระเบียน (Record) ข้อมูลใหม่ในตาราง เช่น คำสั่ง Insert เป็นต้น คำสั่งในการแก้ไขข้อมูลในสดมภ์ (Column) เช่น คำสั่ง Update เป็นต้น คำสั่งในการลบข้อมูลในตาราง ระเบียน (Record) เช่น คำสั่ง Delete

ภาษาควบคุมข้อมูล (Data Control Language : DCL) ใช้สำหรับควบคุม ฐานข้อมูล การเข้าถึงข้อมูล ตัวอย่างเช่น คำสั่งยกเลิกการเข้าถึง ฐานข้อมูล ได้แก่ คำสั่ง Revoke คำสั่งควบคุมการเข้าถึงฐานข้อมูล ได้แก่ Grant

4. การเชื่อมต่อระหว่างตารางต่างๆ ภายในฐานข้อมูล ซึ่งผู้ที่ใช้จะไม่มี โอกาสได้รู้ได้เห็น เพราะระบบจะปฏิบัติการให้อัตโนมัติ โดยจะเลือกกลยุทธ์ที่ดีที่สุด และได้ผลที่ดีที่สุด

5. การประมวลผลฐานข้อมูลแบบสัมพันธ์ จะเป็นแบบกลุ่มข้อมูล (Set Oriented) โดยการถามหรือค้นหาข้อมูลจะได้คำตอบเป็นชุดหรือหลายๆ เซต ระเบียบวน กลับมา ต่างกับการจัดการฐานข้อมูล (DBMS) ในอดีตจะให้คำตอบครั้งละระเบียบวนแล้วมีการวนรอบเรียบระเบียบวนต่อไปในฐานข้อมูลแบบสัมพันธ์ จะไม่มีการวนรอบ แต่จะมีการใช้งานผ่านตัวกระทำสัมพันธ์พื้นฐาน เช่น Select, Join, Project, Divide กับตัวกระทำทางเซต เช่น การตัดกัน (Intersect) การผนวก (Union) การทำผลคูณ (Product) การทำผลต่าง (Difference)

6. ข้อมูลเกี่ยวกับระบบ ตั้งแต่โครงสร้างฐานข้อมูล สิทธิของการใช้ จะต้องเก็บบันทึกในรูปแบบตาราง เมื่อฐานข้อมูลมีการเปลี่ยนแปลง โครงสร้าง ตารางระบบนี้จะมีการเปลี่ยนแปลงแก้ไขโดยอัตโนมัติ ซึ่งจะเรียกตารางนี้ว่า พจนานุกรม (Data Dictionary) การค้นหรือการขอข้อมูลในพจนานุกรมตารางข้อมูล จะต้องสั่งด้วยภาษาฐานข้อมูลชุดเดิม

7. ฐานข้อมูลแบบสัมพันธ์และแบบโครงสร้าง สามารถทำการเปลี่ยนแปลง แก้ไขได้ โดยสามารถเปลี่ยนแปลงที่เค้าร่างย่อยของ ส่วนการมองของผู้ใช้หรือจะเปลี่ยนค่าพร้อมกัน ในหลายแถวหรือหลายสดมภ์

8. การเปลี่ยนแปลงฐานข้อมูลในระดับต่ำ ไม่มีผลกระทบต่อส่วนในระดับสูงกว่าหรือชุดคำสั่งที่ใช้งาน ไม่ว่าจะเปลี่ยนแปลงในระดับภายในฐานข้อมูลหรือในระดับเชิงแนวคิด

9. ระบบฐานข้อมูลควรจะต้องเป็นอิสระ ไม่ขึ้นกับการจัดเก็บตามข้อกำหนด และความถูกต้องของข้อมูล ไม่ขึ้นกับการควบคุมของภาษาระดับต่ำ และไม่ขึ้นกับการกระจายของข้อมูลที่อาจจะแยกกันอยู่หลายๆ แห่ง

คุณสมบัติเหล่านี้ เป็นข้อดีของระบบฐานข้อมูลแบบสัมพันธ์ในการสร้างฐานข้อมูล เพื่อลดความซ้ำซ้อนของข้อมูลในฐานข้อมูล และยังสามารถหลีกเลี่ยงความขัดแย้งของข้อมูล และสามารถปรับปรุง และพัฒนาระบบฐานข้อมูล ได้สะดวกต่อไปในอนาคต โดยที่จะไม่ส่งผลกระทบต่อคำสั่งที่ใช้งานและยังมีระบบจัดการด้านความปลอดภัยของข้อมูล เพื่อให้ผู้ใช้สามารถดำเนินการจัดการกับข้อมูลได้ตามสิทธิของผู้ใช้แต่ละคน

### 2.8.5 องค์ประกอบของระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล[9] (Database System) จะประกอบไปด้วย ฐานข้อมูล (Database) ระบบจัดการฐานข้อมูล (Database Management System หรือ DBMS) และ Data Dictionary โดยที่ฐานข้อมูลจะเป็นที่จัดเก็บข้อมูลต่างๆ ที่เกี่ยวข้องไว้ด้วยกันมี DMBs ทำหน้าที่จัดการกับฐานข้อมูลดังกล่าว และโครงสร้างของฐานข้อมูลจะถูกจัดเก็บไว้ใน Data Dictionary ความสัมพันธ์ระหว่าง “ฐานข้อมูล” และ “ระบบฐานข้อมูล” มีความแตกต่างกันอย่างไร สำหรับฐานข้อมูลจะเป็นเพียงส่วนหนึ่งของระบบฐานข้อมูลเท่านั้น ดังนั้นหากพูดถึงระบบฐานข้อมูล จะต้องประกอบไปด้วย 3 ส่วน คือ ฐานข้อมูล, DBMS และ Data Dictionary แต่สำหรับฐานข้อมูลนั้นจะประกอบไปด้วยตาราง และความสัมพันธ์ระหว่างตาราง

สำหรับ DBMS นับว่าเป็นส่วนสำคัญในระบบฐานข้อมูลเป็นอย่างยิ่ง เปรียบเสมือนผู้จัดการ ฐานข้อมูล ทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้งานกับฐานข้อมูล โดยที่ DBMS จะรับคำสั่งจากผู้ใช้งานหรือจากโปรแกรมต่างๆ หลังจากนั้นจะทำการประมวลผลกับฐานข้อมูลโดยอาศัยโครงสร้างที่จัดเก็บไว้ใน Data Dictionary (โครงสร้างของฐานข้อมูลเหล่านี้จะเรียกว่า Meta Data) และทำหน้าที่ส่งผลลัพธ์ที่ได้กลับคืนไปยังผู้ใช้งานหรือโปรแกรม โดยที่ผู้ใช้งานไม่จำเป็นต้องรู้เลยว่า DBMS จัดเก็บข้อมูลอย่างไร มีกลไกในการเข้าถึงหรือค้นหาข้อมูลอย่างไร ขอเพียงรู้คำสั่งที่ต้องการสั่งงานเพื่อให้ได้ผลลัพธ์ที่ต้องการเท่านั้น ที่เหลือเป็นหน้าที่ของ DBMS ในการดึงข้อมูลหรือการประมวลผลต่างๆ ดังนั้น สำหรับผู้ใช้งานจะรู้สึกว่าการใช้งาน DBMS ทำได้ค่อนข้างง่ายดาย เพราะ DBMS จะซ่อนความยุ่งยากในการเข้าถึงข้อมูลไว้เอง สำหรับ DBMS ที่ได้รับความนิยมสูงสุดในปัจจุบันจะเรียกว่า RDBMS (Relational DBMS ) ซึ่ง DBMS นี้จะมีให้เลือกใช้งานมากมาย ทั้งแบบใช้งานคนเดียวหรือหลายคนพร้อมๆ กัน เช่น MS-Access, FoxPro, Paradox เป็นต้น จนถึงในระดับ Server ที่เรียกว่า Database Serve เช่น SQL Server, Oracle, Informix, Sybase เป็นต้น

## 2.8.6 คำศัพท์ที่ควรรู้จักเกี่ยวกับฐานข้อมูล

### 1. Table

เป็นที่จัดเก็บข้อมูลของฐานข้อมูล โดยปกติฐานข้อมูลหนึ่งจะประกอบไปด้วยข้อมูลหลายๆ ตารางรวมกัน โดยที่ตารางข้อมูลจะประกอบไปด้วยเรคคอร์ด (Record) และ ฟیلด์ (Field)

### 2. Query

เป็นการเรียกค้นข้อมูลที่ต้องการ โดย จะใช้ SQL เป็นภาษาในการคิวรี ซึ่งข้อมูลที่ได้จากการคิวรี โดยภาษา SQL จะเป็น เรคคอร์ดเซต (Recordset) ซึ่งเป็นกลุ่มข้อมูลที่ได้จากการทำคิวรี สำหรับเรคคอร์ดเซตที่ได้สามารถนำไปประมวลผล ต่อไปได้

### 3. Index

เป็นการทำดัชนีของข้อมูลในตาราง เพื่อให้การค้นหาข้อมูลทำได้รวดเร็ว โดยที่อินเด็กซ์สามารถประกอบไปด้วยหลาย ฟیلด์รวมกันหรือเป็นฟیلด์เดี่ยวก็ได้

### 4. Primary Key

เป็นค่าหลักของเรคคอร์ดในตารางเพื่อใช้ในการอ้างเพื่อเข้าถึงข้อมูล ซึ่งค่าของ Primary Key ใน เรคคอร์ด จะต้องไม่ซ้ำกันเรคคอร์ดอื่นในตาราง (มีคุณสมบัติ Uniqueness) โดยจะใช้ฟیلด์ที่ทำอินเด็กซ์มาเป็น Primary Key

### 5. Foreign Key

เป็นฟیلด์ที่อยู่ในตารางหนึ่ง เพื่อใช้อ้างอิงถึงข้อมูลที่อยู่ในอีกตารางหนึ่ง ซึ่งฟیلด์ที่ใช้เป็น Foreign Key ควรจะเป็น Primary Key ของอีกตารางหนึ่งที่มีความสัมพันธ์

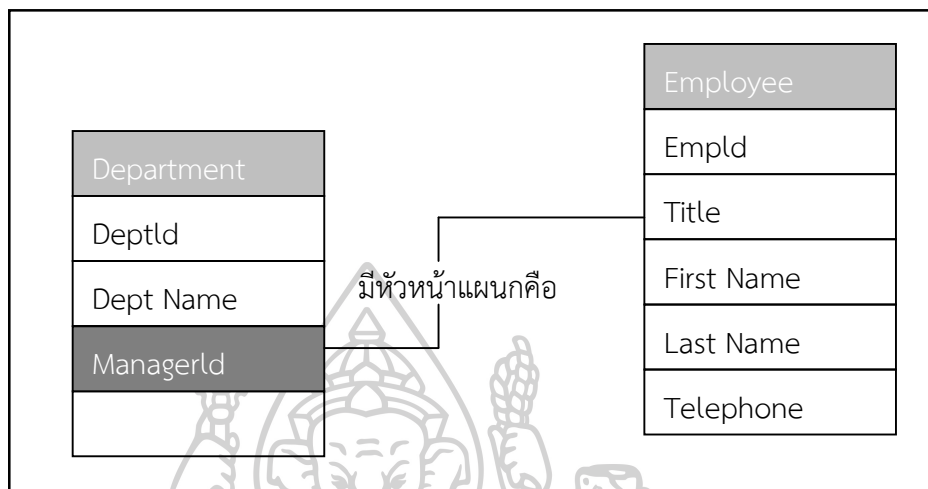
### 6. Relationship

เป็นความสัมพันธ์ระหว่างตารางข้อมูล ซึ่งประกอบไปด้วยความสัมพันธ์ 3 แบบคือ 1 : 1 (One-To-One) 1 : M (One-To-Many) และ M : N (Many-To-Many) ทุกตารางในฐานข้อมูลจะต้องมีฟیلด์ เพื่อเชื่อมความสัมพันธ์ระหว่างตาราง และเรียกฟیلด์นี้ว่า Foreign Key



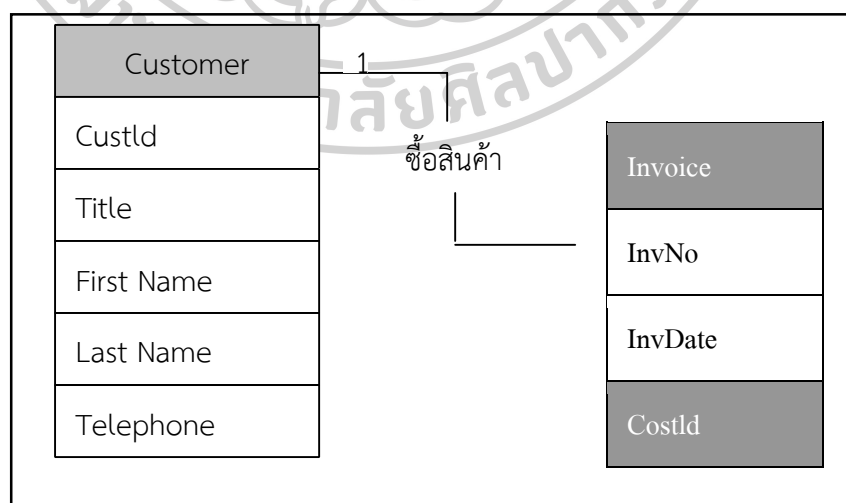
2.8.7 รูปแบบความสัมพันธ์แบบต่างๆ

1. ตัวอย่างข้อมูลในแต่ละตาราง ความสัมพันธ์แบบ 1 : 1 เป็นความสัมพันธ์ที่ในหนึ่งเรคคอร์ดของตารางหนึ่ง มีความสัมพันธ์กับอีกเรคคอร์ดของตารางอื่น



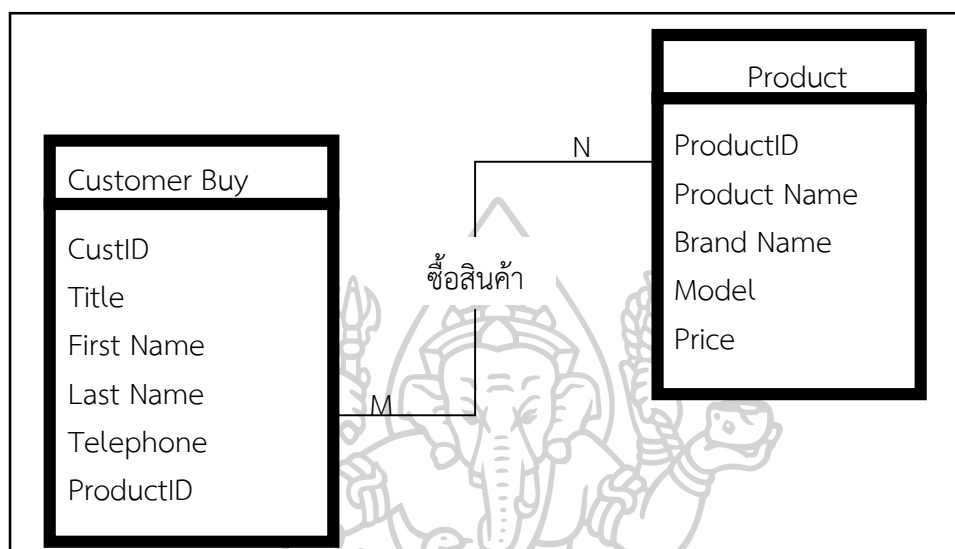
ภาพที่ 13 แสดงความสัมพันธ์ 1 : 1

2. ตัวอย่างข้อมูลความสัมพันธ์แบบ 1 : M เป็นความสัมพันธ์ที่ในหนึ่งเรคคอร์ดของตารางหนึ่งมีความสัมพันธ์ กับอีกหนึ่งหรือหลายเรคคอร์ดของตารางอื่น สำหรับลูกค้าหนึ่งคนสามารถตั้งซื้อสินค้าได้หลายครั้ง และใบกำกับสินค้าหนึ่งใบก็สามารถมีลูกค้าได้เพียงคนเดียวเท่านั้น เช่น นายสมศักดิ์ ทองแก้ว ตั้งซื้อสินค้าจากบริษัททั้งสิ้น 2 ครั้ง ดังนั้นความสัมพันธ์ระหว่างตารางลูกค้ากับใบกำกับสินค้า จึงถือเป็นความสัมพันธ์แบบ 1 : M



ภาพที่ 14 แสดงความสัมพันธ์ 1 : M

3. ตัวอย่างข้อมูลแบบความสัมพันธ์ M : N เป็นความสัมพันธ์ที่มีข้อมูลหนึ่งหรือหลายเรคคอร์ดในตารางหนึ่ง มีความสัมพันธ์กับหนึ่งหรือหลายเรคคอร์ดในตารางอื่น เช่นลูกค้าคนหนึ่งสามารถซื้อสินค้าได้หลายรายการ และสินค้าหนึ่งรายการก็สามารถถูกลูกค้าหลายคนซื้อ ได้เช่นกัน โดยความสัมพันธ์แบบนี้จะเรียกว่าความสัมพันธ์แบบ M : N



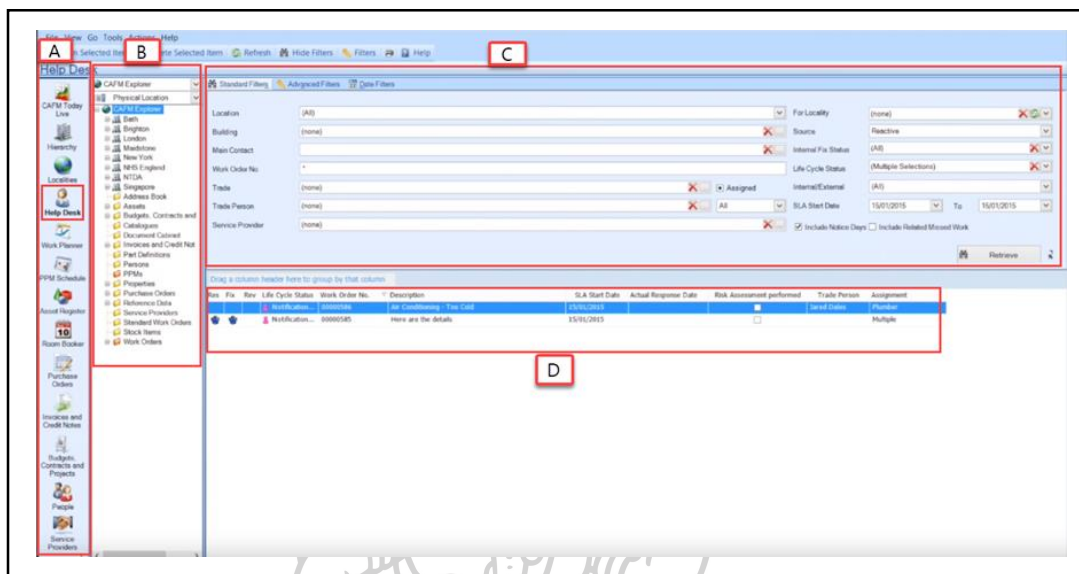
ภาพที่ 15 แสดงความสัมพันธ์แบบ M : N

## 2.9 การศึกษาโปรแกรมที่เกี่ยวข้อง

หากพูดถึง Software สำหรับงานซ่อมบำรุงอาคารในปัจจุบันที่มีอยู่ งานด้านซ่อมบำรุงอาคารจะอยู่ในส่วนประกอบหนึ่งของ Software ประเภท Computerized maintenance management system (CMMS) ซึ่งภายใน Software จะประกอบไปด้วยส่วนต่างๆ หลายส่วนตามความสามารถและราคาของ Software ซึ่งในส่วนการแจ้งซ่อมหรือบำรุงรักษา จะอยู่ในชื่อว่า Workorder ซึ่งรายละเอียดของ Software แต่ละตัวก็จะแตกต่างกันออกไปอีก ผู้ศึกษาได้ทำการค้นคว้าและนำตัวอย่างโปรแกรม Workorder จำนวน 2 โปรแกรมมาทำการศึกษาถึงกระบวนการทำงาน การออกแบบการป้อนข้อมูล รวมถึงการจัดวางหน้าจอต่างๆ ภายในโปรแกรมเพื่อนำมาใช้ในการพัฒนาโปรแกรมซ่อมบำรุงรักษาอาคารต่อไป

## 2.9.1 ขั้นตอนการทำงานของโปรแกรม CAFM Explorer WorkOrder

### 1.โปรแกรม ชื่อ CAFM Explorer (ส่วน Work Order Process) [10]



ภาพที่ 16 แสดงหน้าจอหลักของโปรแกรม [10]

ส่วน A จะเป็นหัวข้องานต่างๆในโปรแกรม ซึ่งในรูปจะอยู่ส่วนงาน HelpDesk

ส่วน B จะเป็นส่วนของ site งานต่างๆ

ส่วน C จะเป็นส่วนสำหรับระบุเงื่อนไขในการค้นหา

ส่วน D จะเป็นส่วนจัดการงาน Work Order ที่มีการส่งเข้ามา

1.ส่วนการแจ้ง WorkOrder โปรแกรม CAFM Explorer จะเปิดให้ผู้แจ้งซ่อมสามารถเข้ามาทาง website ได้จากหน้าจอผู้แจ้งเข้าโปรแกรม โดยการ Login เข้ามาและทำการกรอกข้อมูลต่างๆ ที่ต้องการแจ้งในส่วน B โดยที่ส่วน A จะเป็นส่วนที่แสดงงานแจ้ง WorkOrder ที่ผ่านมา

The screenshot shows the 'New Work Order' form in the CAFM web application. The form is divided into several sections:
 

- Main Contact:** Sally Bates, PB512, sally.bates@cafmssoftware.com, 01273 560000 Ext: 512, Miss, Sales
- Other Contact:** (Empty)
- For Locality:** (CAFM Explorer) (Brighton), Westfield House, UK Sales
- Description:** (Empty)
- Work Order Type:** (Empty)
- Work Order Group:** (Empty)
- Job Details:** (Empty)
- Docket Number:** (Empty)
- Instructions:** (Empty)

 On the left, there is a 'History (2)' section with two entries:
 

- 00000586 - 15/01/2015 17:44
- 00000247 - 15/01/2015 17:43

 On the right, there is a 'Log a Work Order' sidebar with various options like 'Add Main Contact', 'Add Other Contact', etc. At the bottom, there are 'Save' and 'Cancel' buttons. Red boxes labeled 'A' and 'B' highlight the History section and the main form area respectively.

ภาพที่ 17 แสดงหน้าจอการแจ้ง WorkOrder [10]

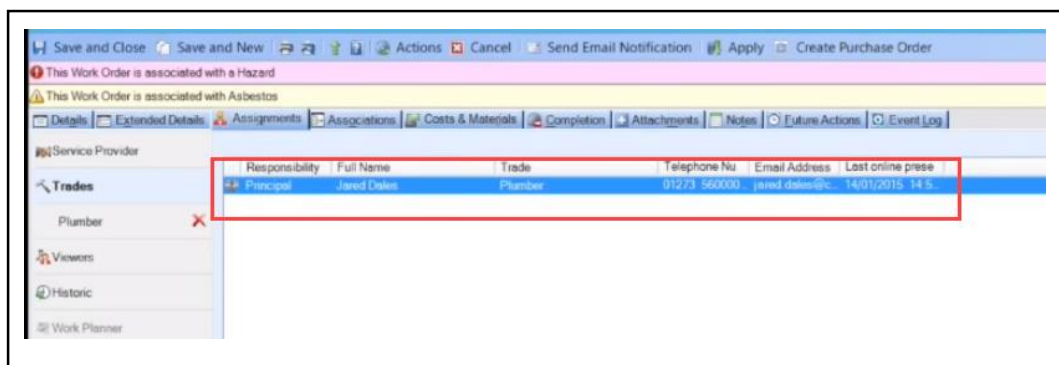
2. หลังจากแจ้ง WorkOrder แล้วที่หน้าจอหลักโปรแกรมจะแสดงงานที่ได้รับขึ้นมา

The screenshot shows the 'Help Desk' interface with a list of Work Orders. The table below is a representation of the data shown in the screenshot:

Res	Fix	Life Cycle Status	Work Order No.	Description	SLA Start Date
		Notification...	00000587	Air Conditioning - Too Cold	15/01/2015
		Notification...	00000585	Here are the details	15/01/2015

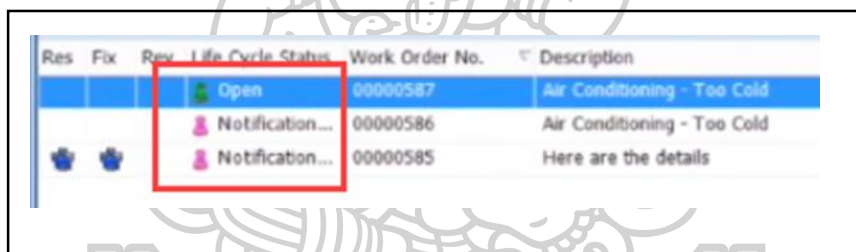
ภาพที่ 18 แสดงงาน WorkOrder ที่ทำการแจ้งเข้ามาในหน้าจอโปรแกรมหลัก [10]

### 3. เป็นขั้นตอนการมอบหมายผู้รับผิดชอบงานที่เกิดขึ้น



ภาพที่ 19 แสดงการเลือกผู้รับผิดชอบงาน [10]

### 4. เป็นการเปิดงาน WorkOrder หรือส่งข้อมูลให้กับผู้ได้รับมอบหมาย



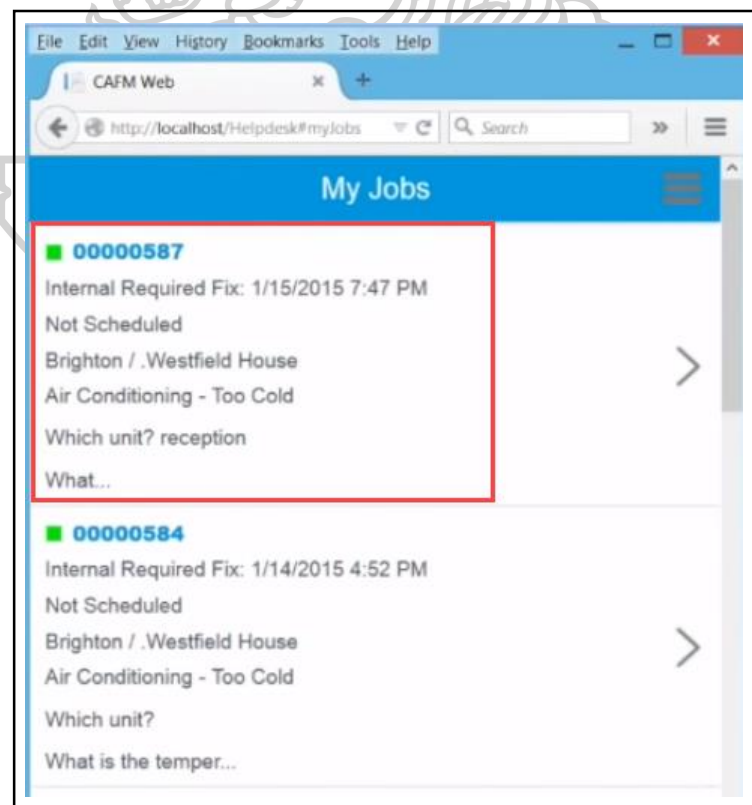
ภาพที่ 20 แสดงสถานะของงานเปลี่ยนเป็น Open [10]

5. หลังจากนั้นผู้ได้รับมอบหมายสามารถเข้า Website เพื่อเข้าดูงานที่ได้รับมอบหมาย



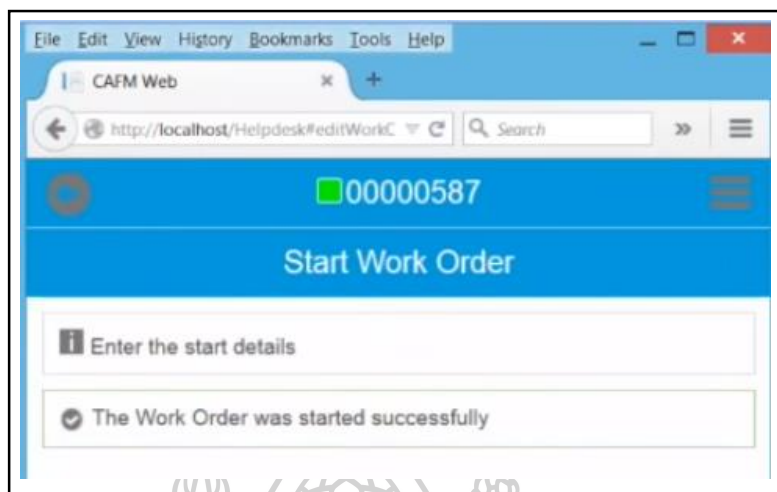
ภาพที่ 21 ผู้ได้รับมอบหมายงานทำการ Login เข้าสู่ระบบ [10]

6. ผู้ได้รับมอบหมายทำการตรวจสอบงานที่ได้รับ



ภาพที่ 22 ส่วนแสดงผลสำหรับช่างที่ได้รับมอบหมายงาน [10]

7.เมื่อผู้ได้รับมอบหมายเริ่มดำเนินการซ่อมต้องทำการกดยืนยันการทำงาน



ภาพที่ 23 แสดงส่วนเริ่มดำเนินการซ่อมช่างยืนยันเริ่มการทำงาน [10]

8.หลังจากการซ่อมแล้วเสร็จผู้ดำเนินการจะต้องรายงานผลการซ่อมกลับไปด้วยเพื่อให้ระบบรับทราบการปิดงาน

ข้อมูลจาก :

<http://www.cafmexplorer.com/cafм-explorer-applications/work-orders/>

ราคาซอฟต์แวร์ [11]

**£5500 per unit**

CAFМ Explorer is a total facilities management product. Functionalities include:

Web, mobile and PC access,

Reactive & planned maintenance,

Health & Safety and Compliance,

Asset tracking of both plant and business critical assets,

Space management, utilisation & planning,

Property Management, leases & tenures,

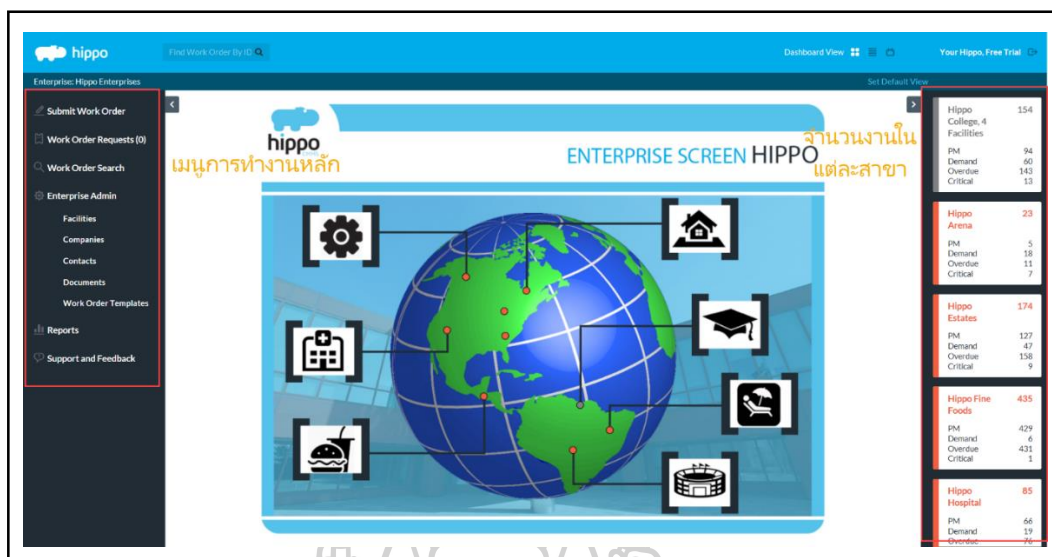
Room Booking,

Cost Control,

Management Information.



## 2.9.2 ขั้นตอนการทำงานของโปรแกรม HIPPO



ภาพที่ 24 หน้าจอหลักของโปรแกรม HIPPO [12]

โดยตัวโปรแกรมจะแบ่งเมนูการทำงานหลักดังนี้

<b>Submit Work Order</b>	ทำหน้าที่รับแจ้งงานต่างๆ
<b>Work Order Requests</b>	หน้าจอจัดการที่รับแจ้งโดยทำการมอบหมายงานและตรวจสอบสถานะและปิดงาน
<b>Work Order Search</b>	หน้าจอกู้คืนงานต่าง
<b>Enterprise Admin</b>	หน้าจอกำหนดค่าต่างๆในโปรแกรม
<b>Reports</b>	แสดงรายงานต่างๆ
<b>Support and Feedback</b>	ส่วนช่วยเหลือในการใช้โปรแกรม

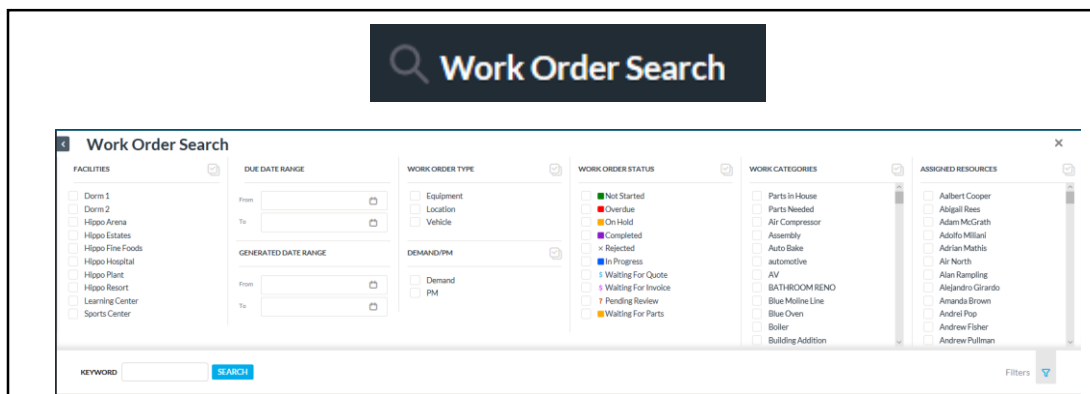
หน้าจอแสดงการรับแจ้งงาน โปรแกรม HIPPO กำหนดให้กรอกข้อมูลในส่วนที่มีสัญลักษณ์ \* ในกรอบสีแดง โดยบางรายการจะเป็นการเลือกข้อมูลที่ทางโปรแกรมกำหนดไว้ หลังจากนั้นจะดึงข้อมูลจากระบบขึ้นมาแสดง

The screenshot shows a web-based form titled "SUBMIT WORK ORDER". The form is organized into several sections:

- Header:** "SUBMIT WORK ORDER" with a help icon and a close button.
- Form Fields:**
  - Work Order Name \***: A text input field with a red border.
  - Work Category**: A dropdown menu with "(None)" selected.
  - Facility \***: A text input field with a red border.
  - Company Category**: A dropdown menu with "All" selected.
  - Type**: Three tabs: "Location" (selected), "Equipment", and "Vehicle".
  - Company**: A dropdown menu with "(None)" selected.
  - Find Location By Floor Plan**: A text input field.
  - Locations \***: A dropdown menu with a red border.
  - Due Date**: A date picker showing "05/12/2018".
  - Priority**: A dropdown menu with "Medium" selected.
  - Critical**: A toggle switch.
  - Description \***: A large text area with a red border.
  - Requested By**: A text input field.
  - Contact Information**: A text input field.
  - Company Contacts**: A text input field.
  - Email Contacts**: A text input field with a "+" button.
  - Send Email to Contacts**: A toggle switch.
  - Email Status Updates**: A toggle switch.
- Buttons:** "CANCEL" and "SUBMIT" buttons at the bottom of the form.
- Footer:** A large dark button with a pencil icon and the text "Submit Work Order".

ภาพที่ 25 หน้าจอแสดงการรับงานแจ้งซ่อมโปรแกรม HIPPO [12]

ส่วนที่ 3 หน้าจอสำหรับค้นหางานที่เข้ามาทั้งหมด



ภาพที่ 26 แสดงการค้นหารายการแจ้งซ่อมที่มีเข้ามาในโปรแกรม [12]

โดยมีหัวข้อที่ใช้ค้นหาดังนี้

**Facilities**

ใช้สถานที่เป็นตัวค้นหา

**Due Date Range**

ใช้ช่วงเวลาในการค้นหา

**Generated Date Range**

ใช้ช่วงเวลาในการแจ้งในการค้นหา

**Work Order Type**

ชนิดของงานแบ่งตามสถานที่หรืออุปกรณ์

**Demand/PM**

ชนิดของงานว่าเป็นร้องขอหรือแบบบำรุงรักษา

**Work Order Status**

สถานะของงานที่อยู่ในระบบ

## Work Categories ประเภทของงาน

<span style="color: green;">■</span> Not Started	Not Started	ยังไม่ดำเนินการ
<span style="color: red;">■</span> Overdue	Overdue	เกินกำหนดการดำเนินงาน
<span style="color: orange;">■</span> On Hold	On Hold	พักงานไว้ชั่วคราว
<span style="color: purple;">■</span> Completed	Reject	ปฏิเสธงานที่ไม่เกี่ยวข้อง
× Rejected	In progress	งานอยู่ระหว่างดำเนินการ
<span style="color: blue;">■</span> In Progress	Waiting For Quote	รอประเมินราคา
<span style="color: lightblue;">\$</span> Waiting For Quote	Waiting For Invoice	รอใบเสนอราคา
<span style="color: magenta;">\$</span> Waiting For Invoice	Pending Review	รอพิจารณา
<span style="color: orange;">?</span> Pending Review	Waiting For Parts	รออะไหล่
<span style="color: yellow;">■</span> Waiting For Parts		

ภาพที่ 27 แสดงรายละเอียดลักษณะงานแจ้งซ่อม [12]

## ส่วนที่ 4 หน้าจอสำหรับรายงาน

ภาพที่ 28 แสดงรายละเอียดส่วนรายงานผลสำหรับเลือกเพื่อสร้างรายงานการแจ้งซ่อม [12]

ที่มา <https://www.hippocmms.com/work-order-software>

<https://www.youtube.com/watch?v=Z7A1IU0STTw>

ราคาซอฟต์แวร์ [12]

Hippo Lite

\$45 USD monthly / user \$55 / CDN monthly Min 3 Users

Work Order Management

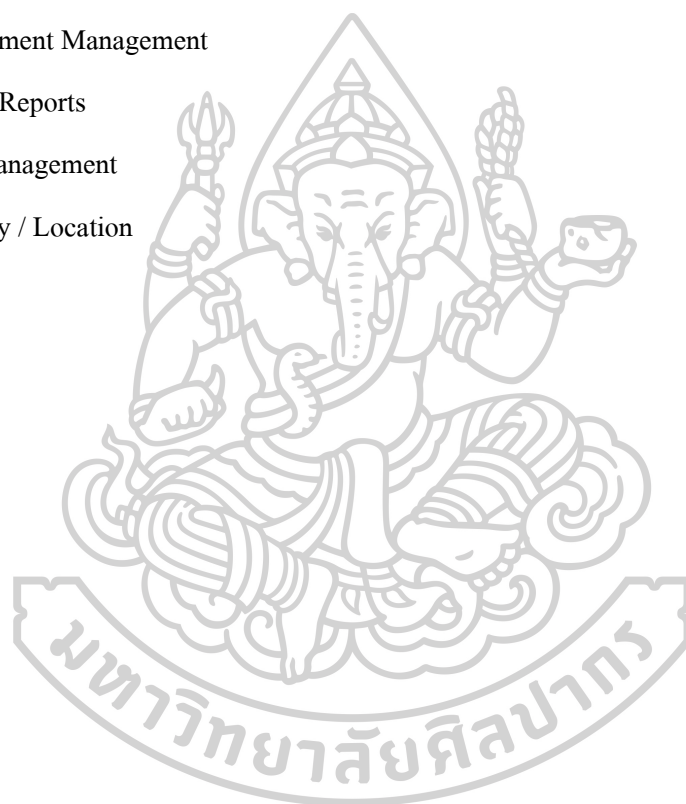
Preventive Maintenance

Asset / Equipment Management

Maintenance Reports

Document Management

Single Facility / Location



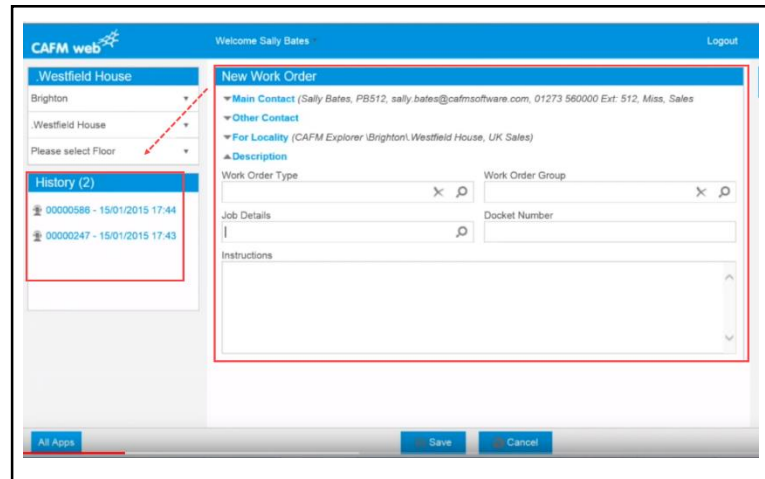
### 2.9.3 สรุปการศึกษาโปรแกรมในกลุ่ม Work Order ที่มีอยู่ในปัจจุบัน

Software ดังกล่าวมีขั้นตอนการทำงานหลักๆ คล้ายกัน โดยเริ่มจากรับคำสั่งเข้ามาในระบบ ไม่ว่าจะเป็นการกรอกข้อมูลในระบบโดยตรงจากผู้ที่อยู่ในหน่วยงานซ่อมบำรุง หรือการกรอกข้อมูลจากผู้อื่นที่ต้องการส่งคำสั่งเข้ามาเพื่อแจ้งซ่อมอุปกรณ์ต่างๆ หลังจากนั้นผู้ดูแลระบบจะทำการมอบหมายงานให้กับช่างไปดำเนินการตามที่ได้รับแจ้ง ซึ่งตัวช่างเองสามารถตรวจสอบงานที่ได้รับมอบหมายได้จากทางเว็บไซต์ เมื่อดำเนินการเสร็จช่างสามารถรายงานผลกลับเข้าระบบได้ผ่านทางเว็บไซต์ ทำให้ทางผู้ดูแลสามารถตรวจสอบการทำงานได้อย่างรวดเร็ว และสั่งปิดงานได้ทันที โดย Software ที่ได้ทำการศึกษา ถูกออกแบบให้สามารถตอบสนองงานต่างๆ ได้อย่างครอบคลุมลักษณะธุรกิจหลากหลายรูปแบบ ทำให้ตัว Software จำเป็นต้องใช้ข้อมูลต่างๆ จำนวนมากทั้งการอ้างอิงหรือการกรอกข้อมูลลงไปในระบบ จึงทำให้โปรแกรมมีขนาดใหญ่ตามไปด้วย และยังต้องใช้เวลาและข้อมูลจำนวนมากในการ Setup โปรแกรมก่อนการใช้งาน มิเช่นนั้นแล้วการทำงานอาจจะไม่สมบูรณ์หรือไม่สามารถใช้งานในบางขั้นตอนได้ ซึ่งเมื่อพิจารณาจากราคา การอบรมการใช้ Software ของผู้ใช้งาน Software ดังกล่าวซึ่งถือว่าเป็นต้นทุนที่สูงมาก อาจไม่เหมาะกับการนำมาใช้ในอาคารขนาดกลางหรือขนาดเล็ก ที่มีเฉพาะหน่วยงานซ่อมบำรุงเพียงอย่างเดียว

### 2.9.4 สรุปข้อดีและการนำมาประยุกต์ใช้

1. software สามารถทำงานได้ผ่านเว็บไซต์ทุกขั้นตอนการทำงาน ไม่ว่าจะเป็นด้าน Input หรือ output ของข้อมูล รวมทั้งจัดเก็บข้อมูลต่างๆ ลงฐานข้อมูล ดังนั้นกระบวนการซ่อมบำรุงรักษาจึงไม่จำเป็นต้องพึ่งพากระดาษทำให้ลดการใช้ทรัพยากรและลดความเสี่ยงด้านข้อมูลสูญหาย

2. การออกแบบหน้าจอภาพโดยแบ่งส่วนการแสดงผลข้อมูลออกเป็นสัดส่วนเพื่อให้การใช้งานโปรแกรมสามารถแสดงผลข้อมูลให้กับผู้ใช้งาน ได้อย่างครบถ้วน ภายในหน้าจอเดียว ความต่อเนื่องในการทำงาน



ภาพที่ 29 แสดงการกรอกข้อมูลการแจ้งซ่อมในหน้าจอหลัก [10]

จากรูปเป็นการกรอกข้อมูลการแจ้งซ่อมในหน้าจอหลักหลังจากกรอกข้อมูลเสร็จโปรแกรมจะไปแสดงเก็บข้อมูลและไปแสดงเฉพาะข้อมูลที่เป็นในหน้าจอเล็ก ทำให้ผู้ใช้สามารถกรอกข้อมูลได้ต่อโดยไม่ต้องเปลี่ยนหน้าจอ

3.การออกแบบหน้าจอให้แสดงข้อมูลในรูปแบบตาราง เพื่อให้สามารถประหยัดพื้นที่ในการแสดงผลในกรณีที่มีข้อมูลปริมาณมากๆ และยังสามารถแสดงสถานะที่แตกต่างกันได้ในแต่ละงานอีกด้วย

Res	Fix	Rev	Life Cycle Status	Work Order No.	Description
			Open	00000587	Air Conditioning - Too Cold
			Notification...	00000586	Air Conditioning - Too Cold
			Notification...	00000585	Here are the details

ภาพที่ 30 หน้าจอแสดงข้อมูลในรูปแบบตาราง [10]

4.การใช้สีในการสื่อถึงความแตกต่างของข้อมูล ทำให้ผู้ใช้งานโปรแกรมสังเกตความแตกต่างของสถานะได้อย่างชัดเจนขึ้น

■	Not Started
■	Overdue
■	On Hold
■	Completed
×	Rejected
■	In Progress
⌘	Waiting For Quote
⌘	Waiting For Invoice
?	Pending Review
■	Waiting For Parts

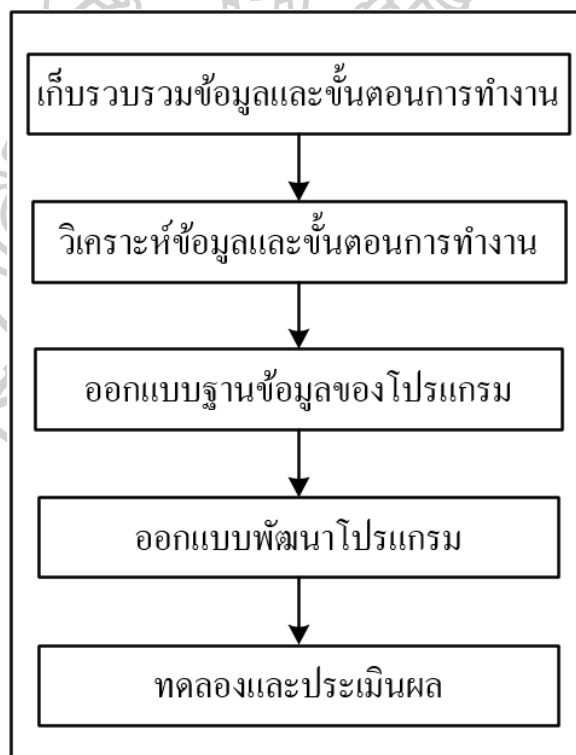
ภาพที่ 31 แสดงการใช้สีในการสื่อถึงความแตกต่างของข้อมูล [12]



### บทที่ 3

#### กระบวนการพัฒนาโปรแกรม

ในบทที่ 3 นี้จะกล่าวถึงกระบวนการพัฒนาโปรแกรมบริหารจัดการบำรุงรักษาอาคาร ซึ่งการพัฒนาโปรแกรมบริหารจัดการบำรุงรักษาอาคาร (Facilities Operations and Maintenance Software) ขึ้นมาเพื่อที่จะช่วยจัดการงานซ่อมบำรุงให้มีประสิทธิภาพมากยิ่งขึ้น และยังช่วยเก็บรวบรวมข้อมูลอุปกรณ์ต่างๆของอาคารไม่ให้เกิดการสูญหาย ผู้ศึกษาจึงทำการศึกษางานซ่อมบำรุงอาคาร โดยเริ่มจากเก็บรวบรวมข้อมูลต่างๆที่เกี่ยวข้องกับการซ่อมบำรุงรักษาอาคาร และทำการเก็บข้อมูลกระบวนการทำงานการซ่อมบำรุงจากหน่วยงานที่ทำหน้าที่ซ่อมบำรุงรักษาอาคาร ซึ่งเป็นกลุ่มเป้าหมายหลักที่จะเป็นผู้ใช้งานโปรแกรมที่พัฒนาขึ้น เพื่อให้ได้โปรแกรมที่สามารถใช้งานได้เหมาะสมกับงานซ่อมบำรุงรักษาอาคาร โดยผู้ศึกษาได้ทำการสรุปเป็นแผนภูมิของขั้นตอนการดำเนินการดังต่อไปนี้



ภาพที่ 32 แผนภูมิแสดงวิธีดำเนินการศึกษา

### 3.1 ศึกษาการทำงานของหน่วยงานซ่อมบำรุงรักษาอาคาร

ในการค้นหาข้อมูลและทำการศึกษาหน่วยงานซ่อมบำรุงรักษาอาคารจะแบ่งเป็น 2 ลักษณะด้วยกัน เป็นการจ้างหน่วยงานภายนอกเข้ามาทำการบำรุงรักษาโดยไม่มีช่างประจำอาคารซึ่งลักษณะนี้จะพบในอาคารขนาดเล็กหรืออาคารที่อยู่อาศัย และอีกลักษณะคือมีหน่วยงานซ่อมบำรุงเป็นหน่วยงานเดียวกันกับทางเจ้าของอาคารมีช่างประจำอาคาร โดยจะพบในอาคารที่ประเภทโรงพยาบาล สำนักงานให้เช่า มหาวิทยาลัย เป็นต้น ซึ่งในการศึกษาครั้งนี้เก็บรวบรวมข้อมูลและวิธีการทำงานจากอาคารที่มีหน่วยงานบำรุงรักษาประจำอาคาร

#### 3.1.1 การเก็บข้อมูลจากการสัมภาษณ์

การสัมภาษณ์ผู้ที่เกี่ยวข้องในหน่วยงานซ่อมบำรุงรักษาประจำอาคาร ซึ่งจะสอบถามถึงขั้นตอนการทำงาน และข้อมูลต่างๆที่เกี่ยวข้องกับการทำงานซ่อมบำรุง โดยแบ่งออกเป็นตำแหน่งต่างๆ ดังนี้ หัวหน้าช่าง ช่าง เจ้าหน้าที่ธุรการประจำแผนกซ่อมบำรุง

### 3.2 วิเคราะห์ข้อมูลจากฝ่ายซ่อมบำรุงรักษาประจำอาคาร

จากการสัมภาษณ์ทำให้ทราบถึงลักษณะงาน ขั้นตอนการทำงานและข้อมูลต่างๆ รวมทั้งปัญหาที่เกิดขึ้นในการทำงาน สามารถสรุปออกมาได้ดังนี้

#### 3.2.1 ลักษณะงานซ่อมบำรุง

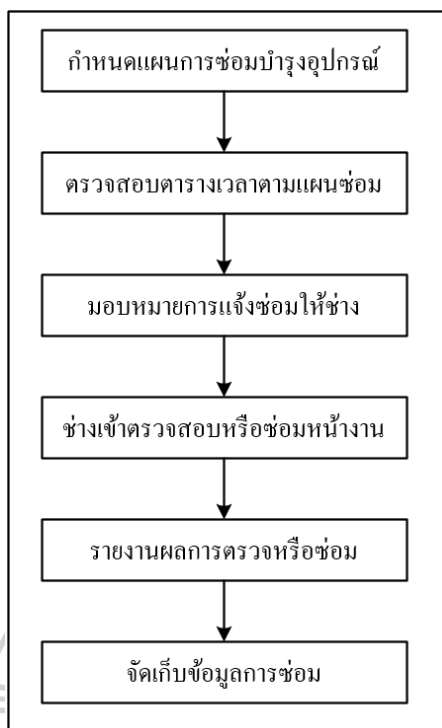
ลักษณะงานซ่อมบำรุงประจำวันจะแบ่งได้เป็น 2 ลักษณะ คือ

3.2.1.1 งานซ่อมบำรุงตามแผนการซ่อมบำรุงที่มีการกำหนดไว้ล่วงหน้า

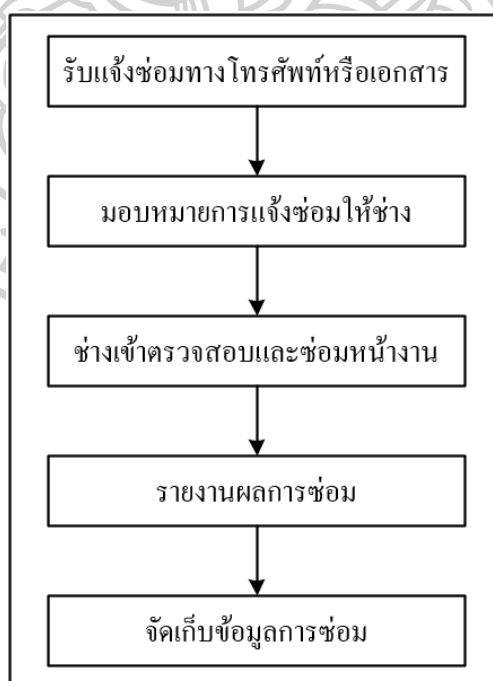
3.2.1.2 งานซ่อมบำรุงที่มีการร้องขอหรือแจ้งซ่อมเข้ามา

#### 3.2.2 ขั้นตอนการทำงาน

เนื่องจากลักษณะงานซ่อมบำรุงประจำวันแบ่งเป็น 2 ประเภทแต่มีขั้นตอนการทำงานในลักษณะเดียวกันเพียงแค่เพิ่มขั้นตอนกำหนดแผนการซ่อมบำรุงอุปกรณ์ขึ้นมา ดังแสดงในแผนภูมิด้านล่าง



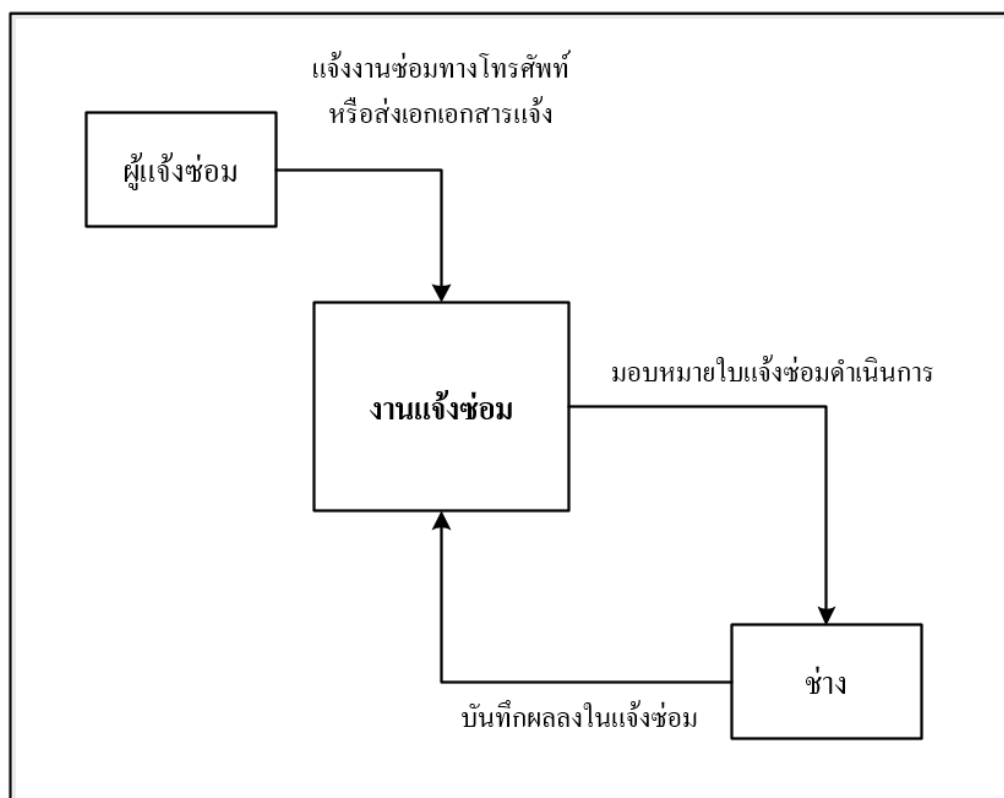
ภาพที่ 33 แผนภูมิแสดงขั้นตอนงานซ่อมบำรุงตามแผนการซ่อมบำรุงที่มีการกำหนดไว้ล่วงหน้า



ภาพที่ 34 แผนภูมิแสดงขั้นตอนงานซ่อมบำรุงที่มีการร้องขอหรือแจ้งซ่อมเข้ามา

### 3.2.3 ระบบงานแจ้งซ่อมในระบบเดิม

ลักษณะของงานแจ้งซ่อมในระบบเดิมการส่งข้อมูลจะอยู่ในรูปแบบการแจ้งผ่านทางโทรศัพท์ซึ่งอาจเกิดการผิดพลาดในการสื่อสาร ระหว่างผู้แจ้งกับผู้บันทึกข้อมูล หากข้อมูลที่ได้รับแจ้งไม่ถูกต้อง จะส่งผลถึงผู้ปฏิบัติงานอาจเกิดความล่าช้าหรือปัญหาอาจไม่ได้รับการแก้ไขที่ถูกต้อง ส่วนการแจ้งซ่อมทางเอกสารจะต้องผ่านระบบสารบัญซึ่งมีโอกาสที่เอกสารจะสูญหายก่อนจะถึงผู้รับ และผู้แจ้งไม่สามารถติดตามผลการดำเนินการใดๆ ได้อีกด้วย ดังที่แสดงในรูปที่ 35



ภาพที่ 35 แสดงระบบการทำงานแจ้งซ่อมในระบบเดิม

นอกจากนั้นแล้วในส่วน of ช่างผู้ปฏิบัติงานซ่อม จะได้รับมอบหมายงานในรูปแบบใบแจ้งซ่อมเพื่อเป็นข้อมูลในการดำเนินการซ่อม ซึ่งหากใบแจ้งซ่อมเกิดการชำรุดสูญหายในระหว่างการเดินทางหรือระหว่างการปฏิบัติงาน จะส่งผลกระทบต่อถึงเวลาในการทำงานและการรายงานผลการซ่อมทันที

### 3.2.4 ปัญหาที่พบในกระบวนการของงานซ่อมบำรุง

จากการศึกษางานแจ้งซ่อมในระบบเดิมพบว่ากระบวนการทำงานใช้การบันทึกจัดเก็บข้อมูลลงในกระดาษเป็นหลัก ไม่ว่าจะเป็นรายละเอียดของอุปกรณ์ที่จะต้องบำรุงรักษาตามระยะเวลา หรือการมอบหมายงานแจ้งซ่อมให้กับช่าง ซึ่งทำให้เกิดปัญหาต่างๆตามมาได้เช่นการค้นหาข้อมูลย้อนหลัง บางครั้งนำเอกสารไปหน้างานและเกิดการชำรุดและสูญหาย หรือกรณีใบแจ้งซ่อมหาย และในเรื่องการบันทึกข้อมูลการแจ้งซ่อมรวมทั้งประวัติการซ่อมในแต่ละอุปกรณ์ ทำให้เกิดผลกระทบต่อการทำงานสรุปผลการซ่อมอีกด้วย เนื่องจากไม่มีการบันทึกข้อมูลลงไปในอีกนัยนั้นกระบวนการสั่งงานแบบปัจจุบันยังไม่สามารถติดตามสถานะงานได้อย่างมีประสิทธิภาพ โดยสามารถสรุปปัญหาของระบบเดิมได้ดังนี้

#### 3.2.4.1 ระบบแจ้งซ่อมขาดประสิทธิภาพ

- ด้านการรับข้อมูลทางโทรศัพท์หรือการส่งเอกสารทำให้เกิดการผิดพลาดและล่าช้าในการทำงาน
- ด้านการติดตามสถานะของงานระบบงานเดิมไม่สามารถติดตามงานได้

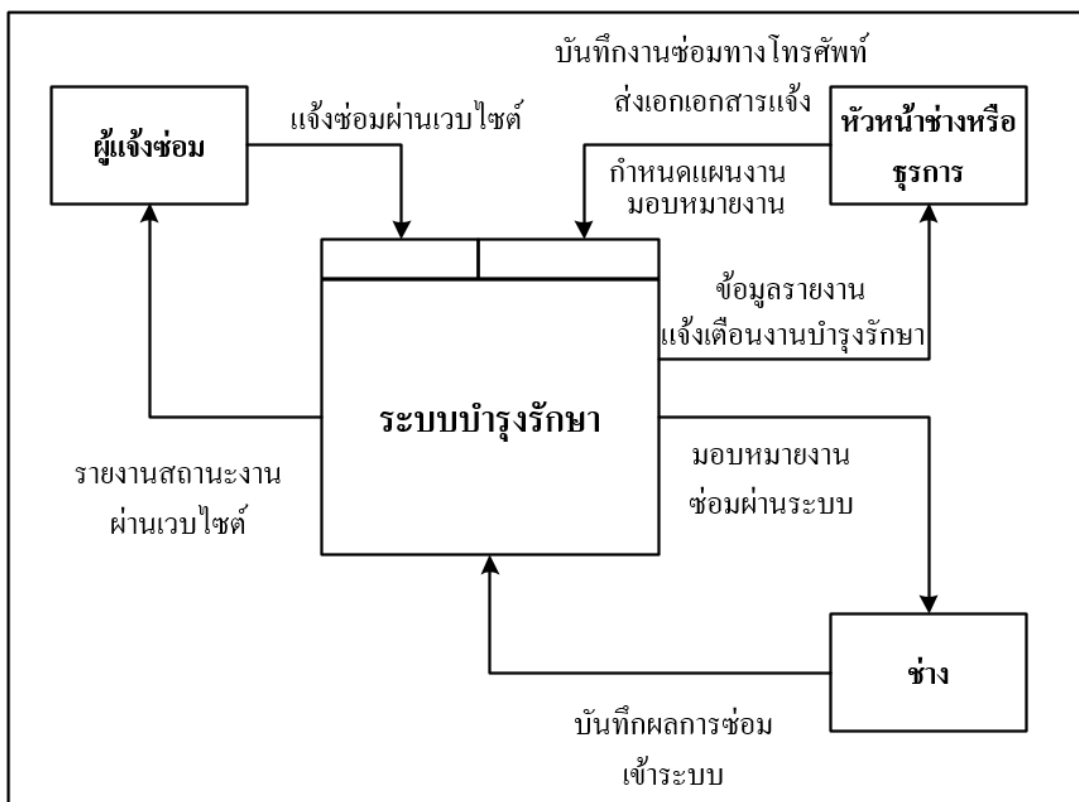
#### 3.2.4.2 ขาดระบบจัดการงานแจ้งซ่อม

- ขาดระบบการบันทึกข้อมูลต่างๆในการแจ้งซ่อม
- ขาดระบบในการตรวจสอบและติดตามสถานะของงาน

### 3.3 การออกแบบระบบงานใหม่

จากการวิเคราะห์ระบบการทำงานเดิมปัญหาหลักคือการบันทึกข้อมูลลงในกระดาษและมีการส่งข้อมูลที่เป็นกระดาษไปยังบุคคลอื่นๆ ทำให้อาจเกิดการสูญหายของข้อมูลหรือผิดพลาดในการบันทึกข้อมูล

ดังนั้นในระบบงานใหม่จึงเลือกการทำงานผ่านเว็บไซต์และเก็บข้อมูลต่างๆไว้ในระบบส่วนกลางเพื่อให้ผู้เกี่ยวข้องสามารถเข้าถึงข้อมูลได้อย่างสะดวกรวดเร็วและเป็นข้อมูลที่มีความถูกต้องและเป็นปัจจุบันมากที่สุด และนอกจากนั้นยังสามารถแสดงสถานะของงานผ่านระบบได้อีกด้วยเพื่อให้ผู้เกี่ยวข้องสามารถติดตามงานได้อย่างรวดเร็วและมีประสิทธิภาพ



ภาพที่ 36 แสดงระบบการทำงานที่ออกแบบใหม่

ในการแจ้งซ่อมจากกระดาษหรือการโทรแจ้งจะถูกแก้ไขด้วยการให้ผู้แจ้งทำการกรอกข้อมูลผ่านทางเว็บไซต์ได้ด้วยตนเองตลอดเวลา ระบบจะจัดเก็บข้อมูลการแจ้งซ่อมไว้ และมีการรายงานสถานะของการดำเนินการของงานอยู่ตลอดเวลา หากผู้แจ้งซ่อมอยากทราบสถานะงานก็สามารถตรวจสอบได้จากทางเว็บไซต์ได้

ส่วนการทำงานของช่างก็สามารถดูและรายงานสถานะและผลการซ่อมผ่านทางเว็บไซต์ได้ทันทีที่ทำงานเสร็จ และทำให้สามารถดูรายการงานแจ้งซ่อมรายการอื่นได้ทันทีโดยไม่ต้องกลับมาที่สำนักงานเพื่อรับใบแจ้งซ่อม

### 3.4 ลักษณะและความสามารถของโปรแกรม

จากการศึกษาและวิเคราะห์ข้อมูลที่ได้มาประกอบกับศึกษาขั้นตอนการทำงานในงานซ่อมบำรุงประจำอาคาร สามารถสรุปความสามารถของโปรแกรมได้ดังนี้  
โปรแกรมแบ่งได้เป็น 4 ส่วนดังนี้

#### 3.4.1 ส่วนงานแจ้งซ่อม

- โปรแกรมสามารถสร้างใบแจ้งซ่อมและมอบหมายงานได้
- โปรแกรมสามารถแสดงสถานะของงานแจ้งซ่อมได้ในทุกขั้นตอนตั้งแต่เริ่มต้นจนจบงาน
- โปรแกรมสามารถรองรับรายงานผลการปฏิบัติงานของช่างพร้อมทั้งเก็บข้อมูลในการซ่อมได้
- โปรแกรมสามารถรองรับการแจ้งซ่อมในรูปแบบที่มีรหัสอุปกรณ์

#### 3.4.2 ส่วนงานแผนบำรุงรักษาแบบป้องกัน

- โปรแกรมสามารถเพิ่มหรือลดรายการข้อมูลอุปกรณ์ประจำอาคารได้โดยผู้ดูแลระบบ
- โปรแกรมสามารถเพิ่มหรือลดแผนการบำรุงรักษาให้กับอุปกรณ์ได้โดยผู้ดูแลระบบ
- โปรแกรมสามารถแจ้งเตือนล่วงหน้าสำหรับงานแผนบำรุงรักษาที่ได้กำหนดไว้ตามแผน

#### 3.4.3 ส่วนผู้ใช้งาน

โปรแกรมมีระบบการแบ่งสิทธิผู้ใช้งาน โดยแบ่งเป็น 3 ผู้ใช้งาน

- ผู้ใช้งานทั่วไปในอาคารซึ่งสามารถทำการกรอกข้อมูลแจ้งซ่อมได้ตัวเอง โดยผ่านการลงทะเบียน
- ผู้ใช้งานในส่วน of ช่างผู้ได้รับมอบหมายงานซ่อมสามารถดูรายการซ่อมและรายงานผลการซ่อมกลับเข้าระบบได้
- ผู้ดูแลระบบหรือหัวหน้าสามารถใช้งานในด้านดังนี้
  1. ออกใบแจ้งซ่อมในกรณีที่มีการแจ้งซ่อมผ่านทางโทรศัพท์หรือส่งเป็นกระดาษ
  2. มอบหมายงานซ่อมให้กับช่าง
  3. เพิ่มหรือแก้ไขข้อมูลของช่างผู้ปฏิบัติงานได้
  4. ออกรายงานในส่วนงบประมาณที่ใช้ในการซ่อมบำรุง

#### 3.4.4 ส่วนจัดเก็บข้อมูลอุปกรณ์

- โปรแกรมสามารถเพิ่มจำนวนข้อมูลอุปกรณ์ที่จะทำการบำรุงรักษาได้



### 3.5 การออกแบบและพัฒนาโปรแกรม

จากข้อมูลต่างๆ ที่ได้ศึกษามาทั้งหมดได้นำมาออกแบบเป็นโปรแกรมที่มีความสามารถรองรับการใช้งานกับหน่วยงานซ่อมบำรุงอาคาร โดยมีส่วนประกอบต่างๆทั้งหมด 5 ส่วนดังนี้

#### 3.5.1 ส่วนแสดงข้อมูลรายละเอียดของงานประจำวัน

ส่วนหน้าแสดงข้อมูลรายละเอียดของงานประจำวันสามารถแสดงรายละเอียดของงานทั้งหมดขึ้นมาทำให้ผู้ใช้งานทำการตรวจสอบหรือสั่งงานได้ภายในหน้าจอเดียว รวมทั้งยังแสดงสถานะโดยรวมของงานทั้งหมด และแจ้งเตือนงานแผนการซ่อมบำรุงที่ได้กำหนดวันเวลาไว้ล่วงหน้าอีกด้วย

ตารางที่ 1 ส่วนหัวข้อแสดงข้อมูลรายละเอียดของงานประจำวัน

ลำดับ	หัวข้อหลัก	รายละเอียด
1.	ผู้เข้าใช้งานระบบ	แสดงชื่อผู้เข้าใช้งานโปรแกรม
2.	งานแจ้งซ่อม	แสดงจำนวนและรายละเอียดของงานแจ้งซ่อมทั้งหมดที่เข้ามา
3.	งานแผนบำรุงรักษา	แสดงจำนวนการแจ้งเตือนงานแผนบำรุงรักษา

#### 3.5.2 ส่วนข้อมูลบุคคลที่ทำงานซ่อมบำรุง(ช่าง)

ส่วนหน้าจอข้อมูลบุคคลที่ทำงานซ่อมบำรุง(ช่าง) จะแบ่งออกเป็น 2 ส่วน ส่วนแรกสามารถบันทึกข้อมูลพื้นฐานของบุคคลไว้ในโปรแกรมและยังสามารถกำหนดตำแหน่งหน้าที่ของบุคคลได้ ส่วนที่สองหน้าจอแสดงรายละเอียดบุคคลแสดงข้อมูลรายละเอียดของบุคคลทั้งหมดที่ได้ทำการบันทึกไว้โดยสามารถแก้ไขข้อมูลได้หรือค้นหาข้อมูลบุคคลจากตำแหน่งหน้าที่ได้

ตารางที่ 2 แสดงหัวข้อส่วนข้อมูลบุคคลที่ทำงานซ่อมบำรุง(ช่าง)

ลำดับ	หัวข้อหลัก	รายละเอียด
1.	เพิ่มข้อมูลบุคคล	จัดเก็บประวัติข้อมูลบุคคล ข้อมูลติดต่อที่ทำงานในหน่วยงานซ่อมบำรุงและกำหนดตำแหน่งหน้าที่
2.	แสดงรายละเอียดบุคคล	แสดงรายละเอียดข้อมูลของบุคคล แก้ไขข้อมูลรายละเอียดบุคคล ค้นหาโดยชื่อหรือตำแหน่งหน้าที่

### 3.5.3 ส่วนข้อมูลอุปกรณ์ประจำอาคาร

หน้าจอส่วนข้อมูลอุปกรณ์ประจำอาคารจะประกอบไปด้วย 2 ส่วน โดยส่วนแรกเป็นการเพิ่มข้อมูลอุปกรณ์โดยการจกเก็บหมายเลขอุปกรณ์เบื้องต้นเพื่อใช้ในการอ้างอิงต่างๆ และยังจกเก็บข้อมูลสถานที่ที่ติดตั้ง ผู้ขายอุปกรณ์อีกด้วย ส่วนที่สองเป็นหน้าจอแสดงรายละเอียดของอุปกรณ์ทั้งหมดที่บันทึกอยู่ในโปรแกรม

ตารางที่ 3 แสดงหัวข้อส่วนข้อมูลอุปกรณ์ประจำอาคาร

ลำดับ	หัวข้อหลัก	รายละเอียด
1.	เพิ่มอุปกรณ์	จกเก็บข้อมูลอุปกรณ์ประจำอาคารไว้ใช้อ้างอิงการแจ้งซ่อมและใช้ในการสร้างแผนการบำรุงรักษา และสามารถสร้างสัญลักษณ์ QR Code ขึ้นมาเพื่อใช้ในการอ้างอิงกับหมายเลขอุปกรณ์
2.	แสดงรายละเอียดอุปกรณ์	แสดงรายละเอียดข้อมูลของอุปกรณ์ ค้นหา แก้ไขข้อมูลรายละเอียดอุปกรณ์

### 3.5.4 ส่วนข้อมูลงานแจ้งซ่อม

หน้าจอส่วนข้อมูลงานแจ้งซ่อมจะประกอบไปด้วย 2 ส่วน โดยส่วนแรกจะแบ่งเป็น 2 ส่วนย่อยในการทำงานเริ่มจาก จกเก็บข้อรายละเอียดการแจ้งซ่อม และมอบหมายงานแจ้งซ่อมให้กับช่าง ส่วนรายละเอียดการแจ้งซ่อมจะเป็นการแสดงผลข้อมูลการแจ้งซ่อมทั้งหมด โดยสามารถสืบค้นข้อมูลพร้อมแสดงรายละเอียดต่างๆ ของงานแจ้งซ่อม

ตารางที่ 4 แสดงหัวข้อส่วนข้อมูลงานแจ้งซ่อม

ลำดับ	หัวข้อหลัก	รายละเอียด
1.	เพิ่มงานแจ้งซ่อม	แบ่งหน้าจอเป็น 2 ส่วนได้แก่ <ul style="list-style-type: none"> <li>- จกเก็บข้อรายละเอียดการแจ้งซ่อม</li> <li>- สามารถออกไปแจ้งซ่อมได้และยังมอบหมายงานแจ้งซ่อมให้กับช่าง</li> </ul>
2.	รายละเอียดการแจ้งซ่อม	แสดงรายละเอียดงานแจ้งซ่อม แสดงสถานะงานแจ้งซ่อม ค้นหา สร้างรายงาน

### 3.5.5 ส่วนแผนงานซ่อมบำรุง

หน้าจอส่วแผนงานซ่อมบำรุงรักษามีการทำงาน 2 ส่วนด้วยกันโดยส่วนแรกเป็นการเพิ่มแผนงานซ่อมบำรุงรักษา ซึ่งจำเป็นต้องเลือกหรือทำการกรอกรายละเอียดของอุปกรณ์เพื่อให้สามารถนำไปใช้ในการสร้างแผนบำรุงรักษาต่อไป และในส่วนที่สองเป็นรายละเอียดแผนซ่อมบำรุงโดยมีการแสดงข้อมูลทั้งตัวแผนงานซ่อมบำรุง รายละเอียดขั้นตอนต่างๆ และในส่วนนี้โปรแกรมยังสามารถสร้างรายงานแผนการบำรุงรักษาอีกด้วย

ตารางที่ 5 แสดงหัวข้อส่วนแผนงานซ่อมบำรุง

ลำดับ	หัวข้อหลัก	รายละเอียด
1.	เพิ่มแผนงานซ่อมบำรุง	หน้าจอบ่งส่วนการทำงานเป็น <ul style="list-style-type: none"> <li>- เลือกอุปกรณ์ที่จะทำการสร้างแผนงานซ่อมบำรุง</li> <li>- สร้างรายละเอียดขั้นตอน</li> <li>- กำหนดผู้รับผิดชอบ</li> <li>- กำหนดความถี่ในงานซ่อมบำรุงรักษา</li> </ul>
2.	รายละเอียดแผนซ่อมบำรุง	แสดงรายละเอียดของแผนซ่อมบำรุง แก้ไขแผนซ่อมบำรุง ค้นหา สร้างรายงาน

### 3.5.6 ส่วนหน้าจอผู้แจ้งซ่อม

ส่วนหน้าจอนี้มีการแบ่งเป็นส่วนแสดงข้อมูลการแจ้งซ่อมที่ผ่านมา โดยมีการแสดงสถานะของงานแจ้งซ่อมไว้ และมีส่วนกรอกข้อมูลการแจ้งซ่อมใหม่ที่ผู้ใช้งานสามารถกรอกรายละเอียดการแจ้งซ่อมไว้และข้อมูลจะถูกส่งเข้าระบบเพื่อดำเนินการต่อไป

ตารางที่ 6 แสดงหัวข้อส่วนหน้าจอผู้แจ้งซ่อม

ลำดับ	หัวข้อหลัก	รายละเอียด
1.	เพิ่มงานแจ้งซ่อม	จัดเก็บข้อรายละเอียดการแจ้งซ่อม
2.	รายการงานซ่อม	แสดงสถานะของงานแจ้งซ่อมที่ได้แจ้งไปแล้ว

### 3.5.7 ส่วนหน้าจอช่าง

ส่วนหน้าจอนี้จะเป็นการแสดงผลการงานแข่งข้อมที่ได้รับมอบหมายงานซ่อมให้กับช่างแต่ละคนโดยจะสามารถทำการยืนยันการดำเนินงานหรือรายงานความก้าวหน้าของงานได้อีกด้วย

ตารางที่ 7 แสดงหัวข้อส่วนหน้าจอช่าง

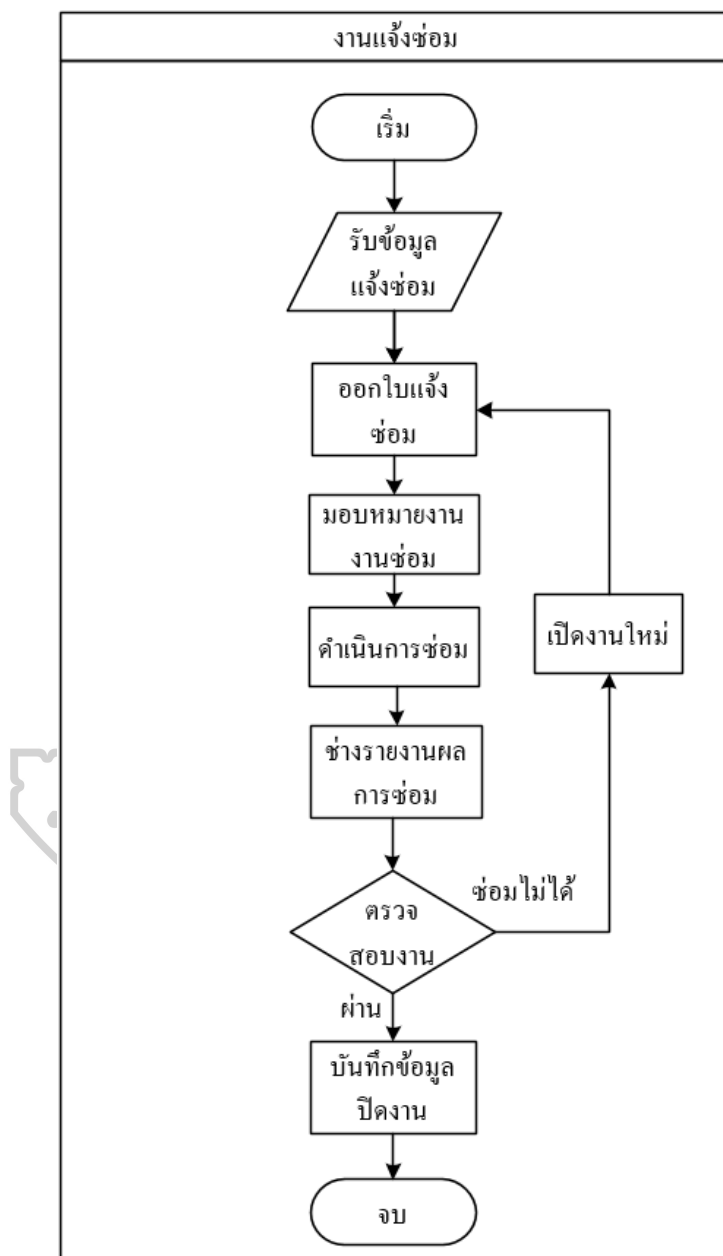
ลำดับ	หัวข้อหลัก	รายละเอียด
1.	งานที่ได้รับมอบหมาย	แสดงงานที่ได้รับมอบหมายให้ดำเนินการทั้งในส่วนการแข่งข้อมและการบำรุงรักษาตามแผน
2.	รายงานการซ่อม	ช่างสามารถรายงานผลการซ่อมได้ในหน้าจอ



### 3.6 ขั้นตอนการทำงาน (Project Flow)

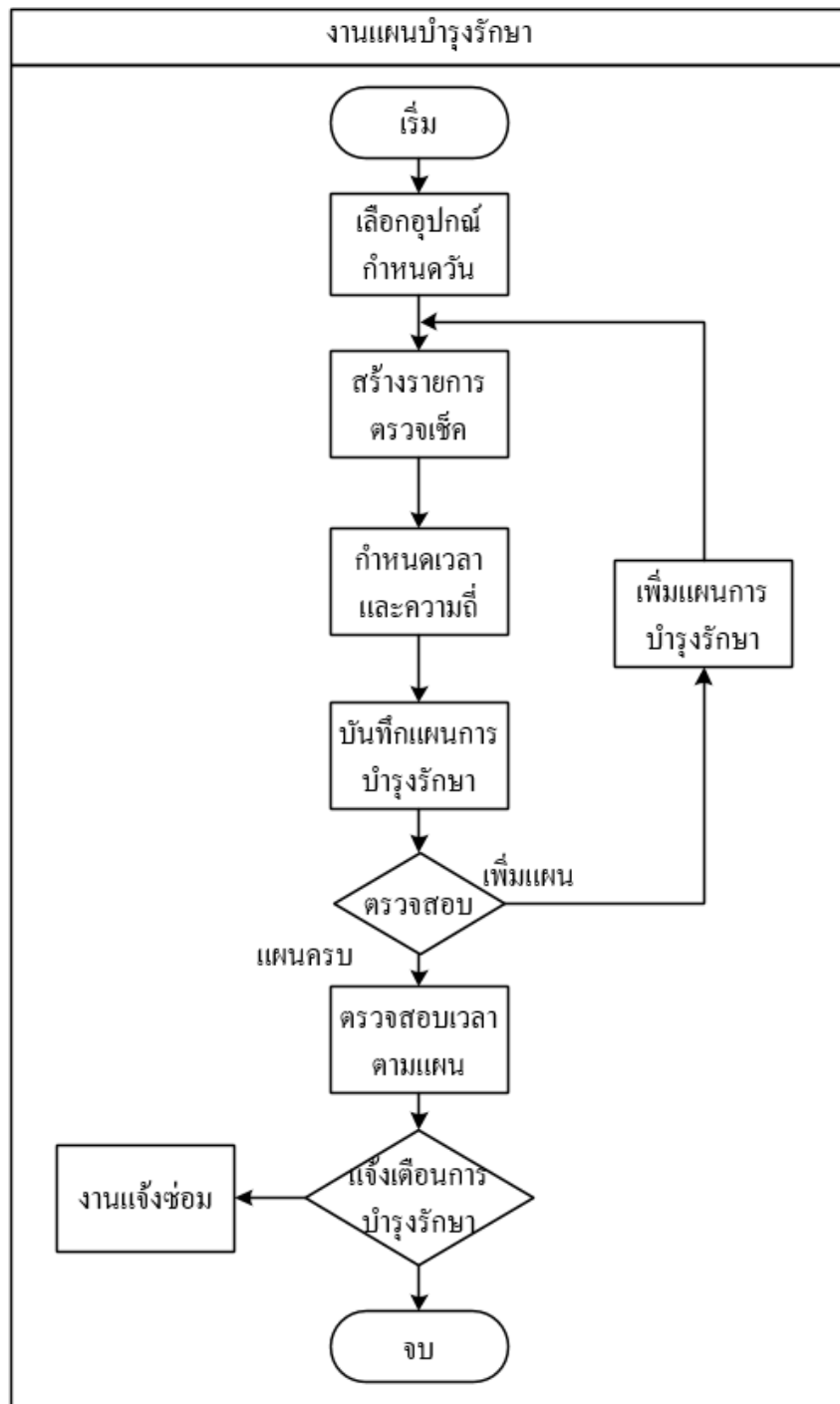
ในส่วนขั้นตอนหลักของ โปรแกรมจะประกอบไปด้วย 2 ส่วนประกอบใหญ่คือ งานแจ้งซ่อมและงานแผนบำรุงรักษา ซึ่งจะประกอบไปด้วยขั้นตอนและรายละเอียดต่างๆ ดังนี้

#### 3.6.1 ขั้นตอนการทำงานแจ้งซ่อม



ภาพที่ 37 Flow Chart แสดงขั้นตอนการทำงานแจ้งซ่อม

## 3.6.2 ขั้นตอนการทำงานแผนบำรุงรักษา



ภาพที่ 38 Flow Chart แสดงขั้นตอนการทำงานบำรุงรักษา

### 3.7 การออกแบบฐานข้อมูล

เพื่อให้โปรแกรมทำงานได้อย่างมีประสิทธิภาพการออกแบบฐานข้อมูลมีความจำเป็นอย่างมากเพราะเป็นส่วนสำคัญในการจัดเก็บข้อมูลต่างๆ อีกทั้งยังช่วยให้โปรแกรมเรียกดูข้อมูลต่างๆได้อย่างมีประสิทธิภาพ โดยการพัฒนาโปรแกรมนี้ได้ออกแบบฐานข้อมูลเป็นตารางไว้ทั้งหมด 8 ตารางเพื่อจัดเก็บข้อมูลต่างๆในโปรแกรมอย่างเป็นระเบียบและมีถูกต้องของข้อมูล โดยกลุ่มของฐานข้อมูลจะถูกแบ่งเป็น 3 กลุ่มคือ กลุ่มฐานข้อมูลบุคคล กลุ่มฐานข้อมูลการแจ้งซ่อม กลุ่มฐานข้อมูลแผนบำรุงรักษา หลังจากได้ตารางข้อมูลแล้วจึงทำการออกแบบความสัมพันธ์ของข้อมูล (ER Diagram)

#### 3.7.1 ตารางฐานข้อมูล

##### 3.7.1.1 กลุ่มฐานข้อมูลบุคคล

ตารางที่ 8 แสดงตารางข้อมูลชื่อ UserInfo

ชื่อตารางข้อมูล	คำอธิบาย
UserInfo	เก็บข้อมูลบุคคลที่เข้าใช้โปรแกรม
Id	PK
Name	ชื่อ
LastName	นามสกุล
Username	ชื่อผู้ใช้
Password	รหัสผ่าน
CreateDate	วันที่สร้าง
IsBackOffice	สถานะเป็นธุรการหรือเป็นผู้แจ้งซ่อม
Building	อาคารที่ทำงาน
Floor	ชั้นที่ทำงาน
Room	ห้องที่ทำงาน
Telephone	เบอร์โทร
Email	อีเมล



ตารางที่ 9 แสดงตารางข้อมูลชื่อ Ref\_Position

ชื่อตารางข้อมูล	คำอธิบาย
Ref_Position	เก็บข้อมูล reference ตำแหน่งของช่าง เช่น ช่างไฟ ช่างปะปา
Id	PK
positionDescription	คำอธิบาย ตำแหน่งช่าง

ตารางที่ 10 แสดงตารางข้อมูลชื่อ TechnicianProfile

ชื่อตารางข้อมูล	คำอธิบาย
TechnicianProfile	เก็บประวัติบุคคล
Id	PK
FirstName	ชื่อจริง
LastName	นามสกุล
Address	ที่อยู่
Telephone	เบอร์โทร
Ref_PositionId	reference บอกว่าช่างคนนี้ตำแหน่งอะไร
CreateDate	วันที่กรอกประวัติช่าง
Username	username ในการเข้าระบบ
Password	password ในการเข้าระบบ
Salary	ค่าแรงต่อชั่วโมง

## 3.7.1.2 กลุ่มฐานข้อมูลการแจ้งซ่อม

ตารางที่ 11 แสดงตารางข้อมูลชื่อ MaintenanceStatus

ชื่อตารางข้อมูล	คำอธิบาย
MaintenanceStatus	สถานะของรายการแจ้งซ่อม
Id	PK
Status	สถานะ

ตารางที่ 12 แสดงตารางข้อมูลชื่อ DefectedStuff

ชื่อตารางข้อมูล	คำอธิบาย
DefectedStuff	อาการชำรุดเบื้องต้น
Id	PK
Detail	หมวดหมู่อุปกรณ์

ตารางที่ 13 แสดงตารางข้อมูลชื่อ MaintenanceJob

ชื่อตารางข้อมูล	คำอธิบาย
MaintenanceJob	เก็บรายการแจ้งซ่อมอุปกรณ์
Id	PK
MaintenanceCode	รหัสการแจ้งซ่อม
MaintenanceRefCode	อ้างอิงรหัสการแจ้งซ่อม ในกรณีที่มีการซ่อมต่อเนื่องมาจากเดิม
CreatorFirstName	ชื่อจริงผู้แจ้งซ่อม
CreatorLastName	นามสกุลผู้แจ้งซ่อม
Telephone	เบอร์โทรศัพท์
HoursForFixed	ชั่วโมงในการซ่อม
Details	รายละเอียดการชำรุด
Ref_EquipmentCode	reference อุปกรณ์ที่แจ้งซ่อม
Ref_DefectedStuffId	reference รายละเอียดการชำรุดเบื้องต้น
Ref_TechnicianId	reference ช่างที่รับงาน
Ref_MaintenanceStatusId	reference สถานะการแจ้งซ่อม
CreateDate	วันที่สร้าง
ModifyDate	วันที่แก้ไขข้อมูล
FollowUp	รายละเอียดการติดตามการซ่อม
CreateBy	สร้างโดย
ModifyBy	แก้ไขโดย
JobType	ประเภทของงาน ปกติหรือด่วน

ตารางที่ 14 แสดงตารางข้อมูลชื่อ EquipmentDetail

ชื่อตารางข้อมูล	คำอธิบาย
EquipmentDetail	เก็บข้อมูลของอุปกรณ์
EquipmentCode	รหัสอุปกรณ์
Name	ชื่ออุปกรณ์
Building	อาคาร
Floor	ชั้น
Room	ห้อง
DetailVendor	รายละเอียดผู้ขายอุปกรณ์
InstallDate	วันที่ติดตั้งอุปกรณ์
ImagePath	path รูปอุปกรณ์
Lifetime	อายุการใช้งาน
Ref_DefectedStuff	reference หมวดหม้ออุปกรณ์

## 3.7.1.3 กลุ่มฐานข้อมูลแผนบำรุงรักษา

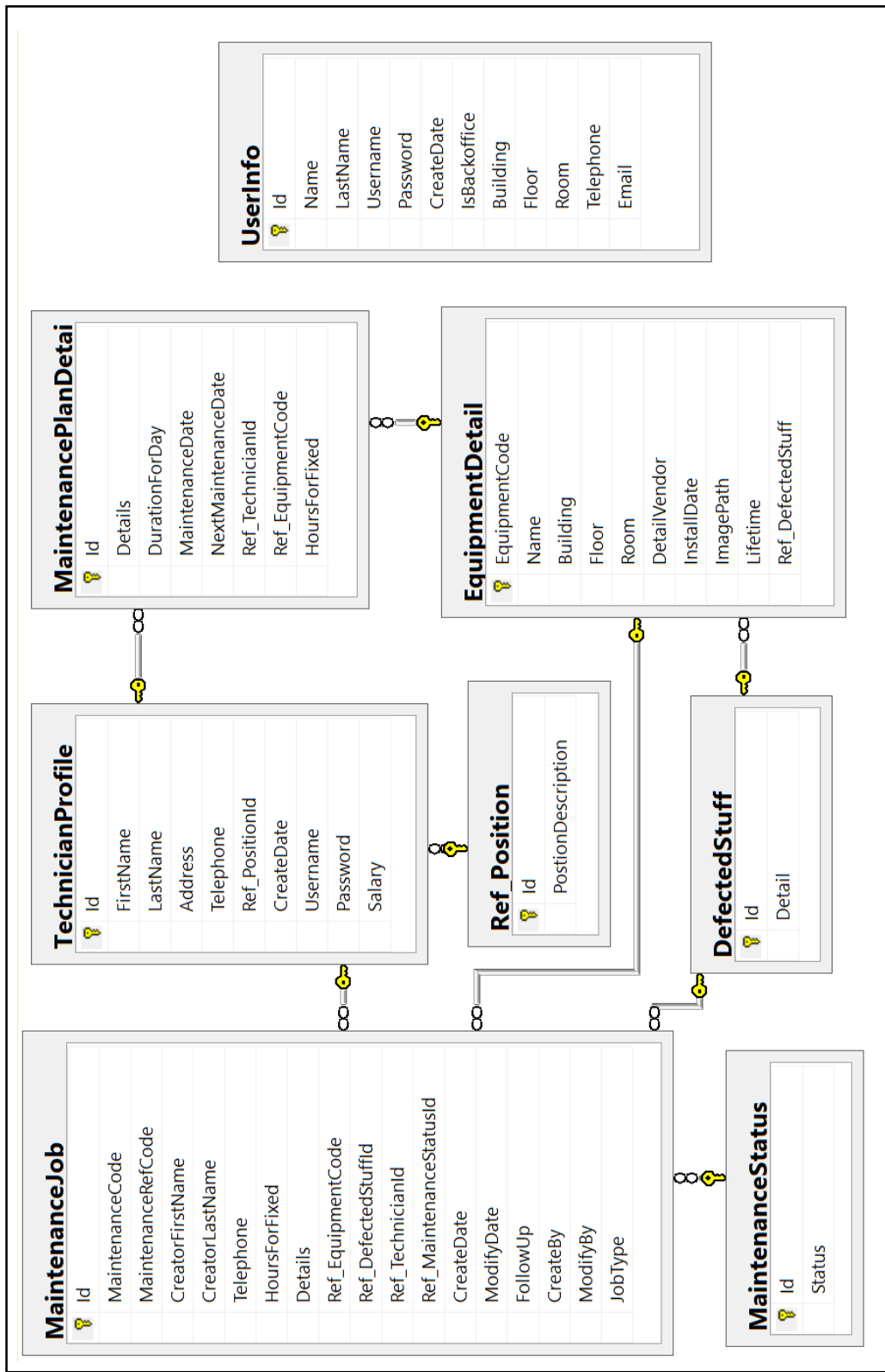
ตารางที่ 15 แสดงตารางข้อมูลชื่อ MaintenancePlanDetail

ชื่อตารางข้อมูล	คำอธิบาย
MaintenancePlanDetail	รายละเอียดแผนการบำรุงรักษา
Id	PK
Details	รายละเอียดในแต่ละขั้นตอน
DurationForDay	ความถี่ในการบำรุงรักษา
Ref_MaintenancePlanMainID	reference แผนหลัก
MaintenanceDate	วันที่บำรุงรักษาล่าสุด
NextMaintenanceDate	วันที่บำรุงรักษาถัดไป

### 3.7.2 ความสัมพันธ์ของข้อมูล (ER Diagram)ฐานข้อมูลระบบโปรแกรมบริหารจัดการ บำรุงรักษาอาคาร

จากข้อมูลตารางทั้ง 3 กลุ่มซึ่งประกอบไปด้วย ฐานข้อมูลบุคคล กลุ่มฐานข้อมูลการแจ้ง  
ซ่อม กลุ่มฐานข้อมูลแผนบำรุงรักษา จึงทำการออกแบบความสัมพันธ์ของข้อมูล (ER Diagram)  
ได้ดังนี้





ภาพที่ 39 ฐานข้อมูลระบบโปรแกรมบริหารจัดการบำรุงรักษาอาคาร (ER Diagram)

## บทที่ 4

### ผลการศึกษา

จากการดำเนินการรวบรวมข้อมูลต่างๆที่จำเป็นต่อการพัฒนาโปรแกรมและได้ออกแบบฐานข้อมูลต่างๆ เสร็จเรียบร้อยแล้วในบทที่ 3 ผู้ศึกษาจึงได้ทำการพัฒนาโปรแกรมบริหารจัดการบำรุงรักษาอาคาร (Facilities Operations and Maintenance Software) ขึ้นมาโดยจะประกอบไปด้วย 2 ส่วนได้แก่

#### 4.1 โปรแกรมที่ใช้ในการพัฒนาระบบ

- 4.1.1 Microsoft Visual Studio 2012 พัฒนาในรูปแบบ ASP.NET ด้วยภาษา C#
- 4.1.2 Microsoft SQL Server 2012 Express
- 4.1.3 MessagingToolkit.QRCode.dll
- 4.1.4 DevExpress.Web.ASPxGridView [13]
- 4.1.5 Internet Browser
- 4.1.6 Google chart tools [14]

#### 4.2 ผังโครงสร้างหน้าจอโปรแกรม

โครงสร้างของโปรแกรมจะอยู่ในรูปแบบโครงสร้างของเว็บไซต์ ซึ่งจะประกอบไปด้วยหน้าต่างๆ จำนวน 14 หน้าหลัก ดังนี้

- 4.2.1 หน้าจอเข้าสู่โปรแกรม (LOGIN) ชื่อไฟล์ Login.aspx
- 4.2.2 หน้าจอลงทะเบียนเข้าใช้งานโปรแกรม(SignUP) ชื่อไฟล์ SignUp.aspx
- 4.2.3 หน้าจอผู้แจ้งซ่อมซึ่งจะประกอบด้วยส่วนการแสดงผลข้อมูลและส่วนการกรอกข้อมูล ชื่อไฟล์ DashBoardUser.aspx , MaintenanceJobUser.aspx
- 4.2.4 หน้าจอการทำงานช่าง ซึ่งจะประกอบไปด้วยส่วนการแสดงผลข้อมูลและส่วนการรายงานผลการซ่อม ชื่อไฟล์ DashBoardTechnician.aspx , MaintenanceJobTechnician.aspx
- 4.2.5 หน้าจอหลักแสดงงานแจ้งซ่อมประจำวัน ชื่อไฟล์ DashBoard.aspx
- 4.2.6 หน้าจอเพิ่มข้อมูลช่าง ชื่อไฟล์ SetupUser.aspx
- 4.2.7 หน้าจอรายละเอียดช่าง ชื่อไฟล์ TechnicianSummary.aspx
- 4.2.8 หน้าจอเพิ่มอุปกรณ์ ชื่อไฟล์ AddEquipment.aspx

4.2.9 หน้าจอรายละเอียดอุปกรณ์ ชื่อไฟล์ EquipmentDetails.aspx

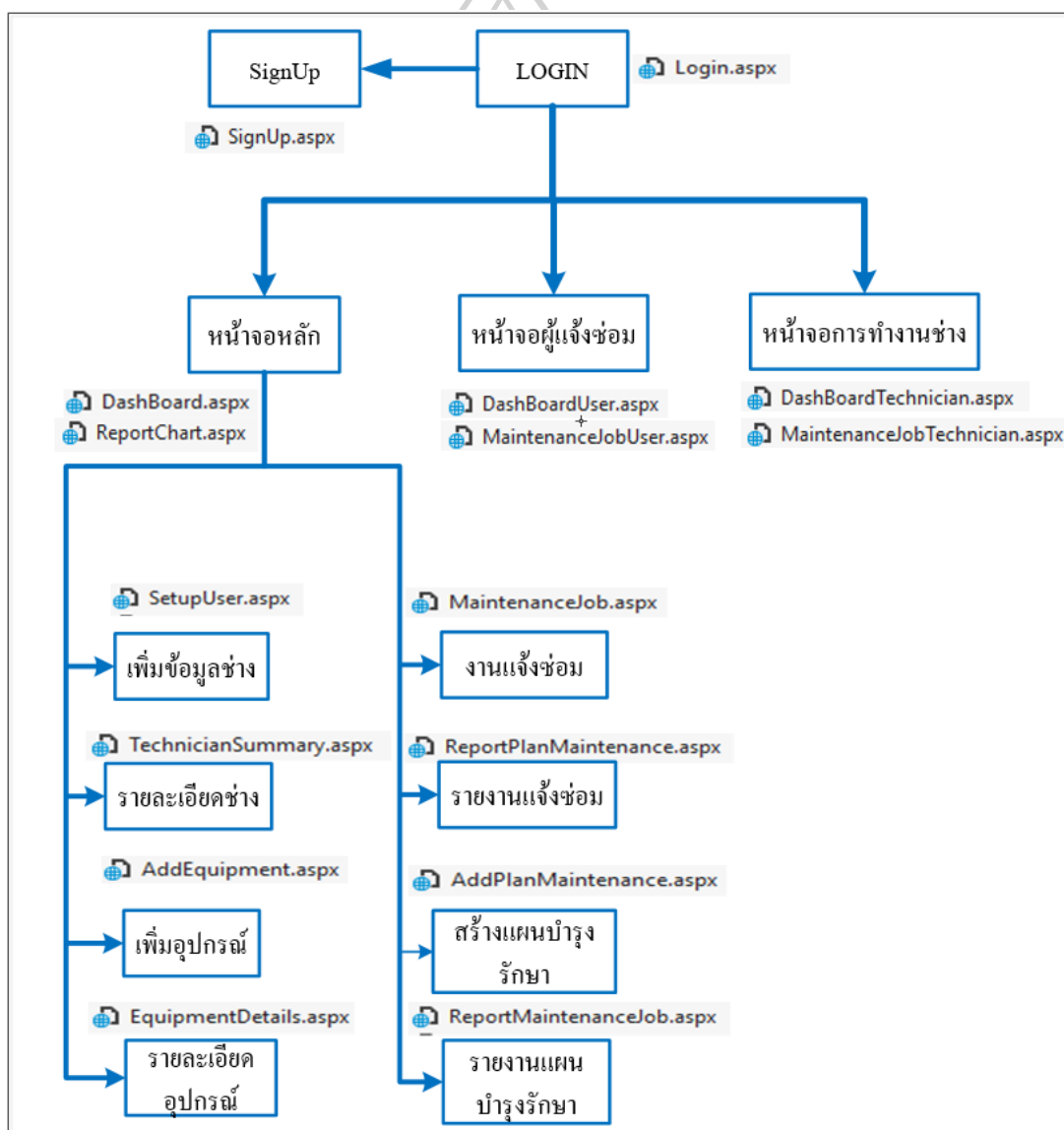
4.2.10 หน้าจองานแจ้งซ่อม ชื่อไฟล์ MaintenanceJob.aspx

4.2.11 หน้าจอรายงานแจ้งซ่อม ชื่อไฟล์ ReportMaintenanceJob.aspx

4.2.12 หน้าจอสร้างแผนบำรุงรักษา ชื่อไฟล์ AddPlanMaintenance.aspx

4.2.13 หน้าจอรายงานแผนบำรุงรักษา ชื่อไฟล์ ReportPlanMaintenance.aspx

4.2.14 หน้าจอแสดงรายงานรวม ชื่อไฟล์ ReportChart.aspx



ภาพที่ 40 แสดงแผนภูมิการเชื่อมโยงหน้าเว็บในโปรแกรม



#### 4.3 ความสามารถของโปรแกรมในการกำหนดสิทธิ

โปรแกรมแบ่งผู้ใช้งานเป็น 3 กลุ่ม

1. Admin เป็นผู้ใช้งานหลักของ โปรแกรมซึ่งอาจจะเป็นหัวหน้าช่างหรือเจ้าหน้าที่ผู้ทำหน้าที่รับแจ้งงาน ซึ่งสิทธิการใช้งานจะมากที่สุดสามารถสร้างและแก้ไขข้อมูลต่างๆในโปรแกรมได้ รวมทั้งการมอบหมายงานต่างๆให้กับช่างด้วย

2. Technician เป็นผู้ใช้งานโปรแกรมในส่วนของการได้รับมอบหมายงานซ่อมที่ได้รับแจ้งเข้ามาในระบบ โดยมีหน้าที่ปฏิบัติตามงานที่ได้รับมอบหมายและทำการรายงานผลการทำงานกลับเข้าระบบ

3. User เป็นผู้ใช้งานอาคารทั่วไปที่ต้องการแจ้งซ่อม โดยสามารถเข้ามากรอกข้อมูลในระบบและสามารถดูรายงานสถานะงานซ่อมที่ได้แจ้งซ่อมไปแล้ว

ตารางที่ 16 แสดงรายละเอียดของสิทธิในกลุ่มผู้ใช้งานต่างๆในระบบ

ลำดับ	การใช้งาน	กลุ่มผู้ใช้		
		Admin	Technician	User
1.	กรอกข้อมูลแจ้งซ่อม	/		/
2.	มอบหมายงานแจ้งซ่อม	/		
3.	เปลี่ยนแปลงสถานะของงานซ่อม	/	/	
4.	เพิ่ม-แก้ไขข้อมูลช่าง	/		
5.	เพิ่ม-แก้ไขข้อมูลอุปกรณ์	/		
6.	สร้างแผนการบำรุงรักษา	/		
7.	ตรวจสอบสถานะของงาน	/	/	/
8.	ข้อมูลรายงานต่างๆ	/		

## 4.4 หน้าจอส่วนติดต่อกับผู้ใช้งานโปรแกรม (User Interface)

### 4.4.1 หน้าจอเริ่มต้น โปรแกรมหน้า Login

**ระบบบริหารและจัดการบำรุงรักษาอาคาร**

**USERNAME**  ← ป้อนชื่อผู้ใช้ (1)

**PASSWORD**  ← ป้อนรหัสผ่าน (2)

เข้าใช้งาน →   ← ลงทะเบียน

ภาพที่ 41 แสดงรูปหน้าจอการเข้าใช้โปรแกรม

การเข้าใช้โปรแกรมจะต้องกรอกชื่อและรหัสของผู้ใช้งานก่อนการเริ่มใช้โปรแกรมหรือสามารถลงทะเบียนเพื่อสร้างรหัสเข้าใช้งานได้ในระดับ user

### 4.4.2 หน้าจอลงทะเบียน(SignUp)

ในหน้าจอนี้จะเป็นส่วนของผู้ใช้งานสิทธิ์ประเภท User ทำการกรอกข้อมูลส่วนตัวต่างๆ ( A )และกรอกข้อมูล Username และ Password ( B ) เพื่อใช้ในการ Login เข้าไปในระบบ

#### สมัครสมาชิก

☰ ข้อมูลสมาชิก

	<input type="text" value="ชื่อ"/>	<input type="text" value="นามสกุล"/>	
A	<input type="text" value="อาคาร"/>	<input type="text" value="ห้อง"/>	
	<input type="text" value="ชั้น"/>	<input type="text" value="เบอร์โทรศัพท์"/>	
	<input type="text" value="Email"/>		
B	<input type="text" value="Username somjai"/>	<input style="background-color: #ffff00;" type="password" value="Password ....."/>	

ภาพที่ 42 แสดงหน้าจอ การลงทะเบียนสมัครสมาชิกเพื่อเข้าใช้งานการแจ้งซ่อม

### 4.4.3 หน้าจอ รายงานรายละเอียดของงาน (Dash Board)

ในหน้าจอนี้จะแบ่งเป็น 2 ส่วน ได้แก่ งานแจ้งซ่อมที่รับเข้ามาในระบบ ( A ) และส่วนงานแผนการบำรุงรักษาที่แจ้งเตือนขึ้นมาล่วงหน้า 7 วันก่อนถึงกำหนดเวลาตามแผน ( B )




ภาพที่ 43 แสดงหน้าจอ รายงานรายละเอียดของงาน (Dashboard)

ในส่วนหน้าจอรายละเอียดงานแจ้งซ่อม วันนี้ จะประกอบไปด้วยเมนูหัวข้อต่างๆ ให้เลือกแสดงผลได้ตามลักษณะของงาน อาการชำรุด สถานะงานซ่อม ( A ) และแสดงรายละเอียดข้อมูลต่างๆของงานแจ้งซ่อมรวมทั้งยังสามารถเข้าไปแก้ไขสถานะงานแจ้งซ่อมได้ในสคริปต์สุดท้าย ( B )



ภาพที่ 44 แสดงหน้าจอ งานแจ้งซ่อมประจำวัน

ในหน้าจอรายละเอียดแผนบำรุงรักษา วันนี้ จะประกอบด้วย 3 ส่วนคือเมนูหัวข้อต่างๆของงานแผนบำรุงรักษา ( A ) และงานแผนบำรุงรักษาที่มีการแจ้งเตือนขึ้นมาเมื่อถึงเวลาที่กำหนด ( B ) และมีปุ่มสร้างใบแจ้งซ่อมสำหรับแผนบำรุงรักษาเพื่อสร้างรายการแจ้งซ่อม ( C )

A	รายละเอียดแผนบำรุงรักษา ล่องหน้า 7 วัน					C
	รหัสอุปกรณ์	ชื่ออุปกรณ์	รายละเอียดและขั้นตอน	ช่างผู้รับผิดชอบ	วันที่ในการดูแล	แก้ไข
B	CCTV-001	CCTV	ตรวจสอบภาพ	ประเสริฐ ชัยชนะ ปรากฏ	29/04/2018	

ภาพที่ 45 แสดงหน้าจองานแผนซ่อมบำรุงประจำวัน

#### 4.4.4. หน้าจอเมนูของโปรแกรม

หน้าจอนี้ประกอบไปด้วย 4 ส่วนดังนี้ ข้อมูลช่าง ( A ) อุปกรณ์ ( B ) งานแจ้งซ่อม ( C ) และแผนซ่อมบำรุง ( D ) ซึ่งในแต่ละส่วนจะประกอบด้วยปุ่มกดเพื่อเปลี่ยนหน้าโปรแกรมไปทำงานตามรายละเอียดต่างๆของงาน

≡ MENU

ข้อมูลช่าง

A

เพิ่มข้อมูลช่าง

รายละเอียดช่าง

อุปกรณ์

B

เพิ่มอุปกรณ์

รายละเอียดอุปกรณ์

งานแจ้งซ่อม

C

เพิ่มงานแจ้งซ่อม

รายละเอียดงานแจ้งซ่อม

แผนซ่อมบำรุง

D

เพิ่มแผนซ่อมบำรุง

รายละเอียดแผนซ่อมบำรุง

ภาพที่ 46 แสดงหน้าจอเมนูของโปรแกรม

#### 4.4.5 หน้าจอเพิ่มข้อมูลช่าง

##### ข้อมูลช่าง

เพิ่มข้อมูลช่าง

A

ชื่อ  นามสกุล

ที่อยู่

เบอร์โทรศัพท์

ตำแหน่ง <--- กรุณาเลือก --- > ค่าแรง / ชม.

B

Username  Password

C

บันทึก กลับ กลับหน้า Dashboard

ภาพที่ 47 แสดงหน้าจอเพิ่มข้อมูลช่าง

หน้าจอนี้ประกอบไปด้วย 3 ส่วนคือส่วนบันทึกข้อมูลพื้นฐานประจำตัวของช่างที่ปฏิบัติงาน (A) และการกรอกข้อมูลของชื่อและรหัสเข้าโปรแกรมของช่าง (B) ส่วนสุดท้ายเป็นปุ่มกดเพื่อบันทึกหรือย้อนกลับไปหน้ารายละเอียดช่าง หรือย้อนกลับไปหน้าจอรายงานรายละเอียดของงาน (C)

#### 4.4.6. หน้าจอข้อมูลรายละเอียดของช่าง

หน้าจอประกอบไปด้วย 3 ส่วน คือส่วนกรอกข้อมูลหรือเลือกข้อมูลที่จะทำการค้นหา(A) การดำเนินการค้นหาจะต้องสั่งงานผ่านปุ่มกดค้นหาจึงจะดำเนินการตามคำสั่ง (B) ข้อมูลที่ทำการสั่งให้ค้นหาแสดงอยู่ด้านล่าง (C) โปรแกรมสามารถจะทำการส่งข้อมูลที่ได้จากการค้นหาออกไปยังโปรแกรม Excel ได้อีกด้วยโดยกดที่ปุ่มบันทึกเป็น Excel (D)

## ข้อมูลรายละเอียดของช่าง

ข้อมูลรายละเอียดของช่าง

A ชื่อ - นามสกุล ช่าง ตำแหน่ง <--- กรุณาเลือก --- >

B ค้นหา บันทึกเป็น Excel เพิ่มข้อมูลช่าง กลับหน้า Dashboard

D

ชื่อ - นามสกุล	ที่อยู่	เบอร์โทรศัพท์	ตำแหน่ง	แก้ไข
Technician01 E-A	เลขที่ 1 หมู่ 3 ตำบลสามพระยา อำเภอชะอำ จังหวัดเพชรบุรี 76120	0-3259-4029	ช่างไฟ	
Technician02 E-A	เลขที่ 1 หมู่ 3 ตำบลสามพระยา อำเภอชะอำ จังหวัดเพชรบุรี 76120	0-3259-4029	ช่างไฟ	
Technician03 P-B	เลขที่ 1 หมู่ 3 ตำบลสามพระยา อำเภอชะอำ จังหวัดเพชรบุรี 76120	0-3259-4029	ช่างประปา	

C

ภาพที่ 48 แสดงหน้าจอข้อมูลรายละเอียดของช่าง

	A	B	C	D	E
1	ชื่อ - นามสกุล	ที่อยู่	เบอร์	ตำแหน่ง	แก้ไข
2	Technician01 E-A	เลขที่ 1 หมู่ 3 ตำบลสามพระยา อำเภอชะอำ จังหวัดเพชรบุรี 76120	0-3259-4029	ช่างไฟ	
3	Technician02 E-A	เลขที่ 1 หมู่ 3 ตำบลสามพระยา อำเภอชะอำ จังหวัดเพชรบุรี 76120	0-3259-4029	ช่างไฟ	
4	Technician03 P-B	เลขที่ 1 หมู่ 3 ตำบลสามพระยา อำเภอชะอำ จังหวัดเพชรบุรี 76120	0-3259-4029	ช่างประปา	
5	Technician04 P-B	เลขที่ 1 หมู่ 3 ตำบลสามพระยา อำเภอชะอำ จังหวัดเพชรบุรี 76120	0-3259-4029	ช่างประปา	
6	Technician05 P-B	เลขที่ 1 หมู่ 3 ตำบลสามพระยา อำเภอชะอำ จังหวัดเพชรบุรี 76120	0-3259-4029	ช่างประปา	

ภาพที่ 49 แสดงหน้าจอข้อมูลรายละเอียดของช่างที่ส่งออกไปยังโปรแกรม Excel

## 4.4.7 หน้าจอเพิ่มอุปกรณ์

หน้าจอในส่วนนี้เป็นการเพิ่มข้อมูลตัวอุปกรณ์เข้าไปในฐานข้อมูลเพื่อใช้อ้างอิงในการแจ้งซ่อมหรือการสร้างแผนการบำรุงรักษา โดยหน้าจอจะแบ่งเป็น 4 ส่วนดังนี้ ส่วนแรกเป็นการกรอกข้อมูลรายละเอียดของอุปกรณ์โดยแยกเป็นรหัสอุปกรณ์และชื่อเรียกอุปกรณ์ อายุการใช้งาน รูปภาพ และหมวดหมู่ของอุปกรณ์ (A) ส่วนที่สองเป็นการระบุสถานที่ตั้งและรายละเอียดข้อมูลผู้ขายอุปกรณ์ดังกล่าว (B) ส่วนที่สามจะเป็นการบันทึกข้อมูลต่างๆ ที่ได้กรอกลงไปโดยการกดปุ่มบันทึก (D) ซึ่งเมื่อกดบันทึกโปรแกรมจะสร้างภาพรหัสอุปกรณ์ขึ้นมาในรูปแบบ QR Code ซึ่งเป็นลักษณะของ ภาพที่มีชื่อรหัสอุปกรณ์ .JPG (C) เมื่อโปรแกรมสร้าง QR Code เสร็จผู้ใช้โปรแกรมสามารถกลับไปหน้าจอหลักได้ในส่วน (D)

**ข้อมูลรายละเอียดอุปกรณ์** A

รหัสอุปกรณ์  ชื่ออุปกรณ์

อายุการใช้งาน

เพิ่มรูปภาพ  No file chosen

หมวดหมู่อุปกรณ์

**สถานที่ติดตั้ง** B

วันที่ติดตั้ง  อาคาร

ชั้น  ห้อง

**รายละเอียดข้อมูลหมายเหตุอุปกรณ์**


C

D

ภาพที่ 50 แสดงหน้าจอเพิ่มข้อมูลอุปกรณ์และแสดงรหัส QR Code



#### 4.4.8 หน้าจอรายละเอียดอุปกรณ์

หน้าจอในส่วนนี้จะเป็นการแสดงข้อมูลของอุปกรณ์ที่ได้ทำการบันทึกเข้ามา โดยสามารถเลือกที่จะค้นหาตามรหัสอุปกรณ์ หรือชื่ออุปกรณ์ได้ในส่วน ( A ) หลังจากกรอกข้อมูลผู้ใช้สามารถสั่งให้ค้นหาได้จากการกดปุ่มค้นหาในส่วน ( B ) เมื่อกดปุ่มค้นหาแล้วจะมีการแสดงข้อมูลตามเงื่อนไขที่ได้ใส่ไว้และยังสามารถแก้ไขข้อมูลที่บันทึกได้อีกด้วย ( C )

#### รายละเอียดอุปกรณ์

☰ ค้นหาข้อมูลอุปกรณ์

A

B

	รหัสอุปกรณ์	ชื่ออุปกรณ์	อาคารที่ตั้ง	ชั้นที่ตั้ง	ห้องที่ตั้ง	แก้ไข
C	A02-No1	Split Type Unit	อาคาร1	1	101	✎
	A02-No10	Split Type Unit	อาคาร1	5	502	✎
	A02-No11	Split Type Unit	1	5	502	✎
	A02-No2	Split Type Unit	อาคาร1	1	102	✎
	A02-No3	Split Type Unit	อาคาร1	2	201	✎

ภาพที่ 51 แสดงหน้าจอรายละเอียดอุปกรณ์



#### 4.4.9 หน้าจองานแจ้งซ่อม

ในหน้าจองานแจ้งซ่อมจะแบ่งส่วนงานเป็น 2 ส่วนด้วยกัน โดยส่วนแรกจะเป็นส่วนหน้าจอการป้อนข้อมูลการแจ้งซ่อม ( A ) ระบบจะทำการสร้างหมายเลขการแจ้งซ่อมขึ้นมาให้โดยอัตโนมัติและไม่สามารถแก้ไขได้ ผู้แจ้งซ่อมจะต้องทำการป้อนรหัสอุปกรณ์เข้าไปในช่องรหัสอุปกรณ์โดยรหัสอุปกรณ์จะถูกบันทึกไว้กับอุปกรณ์ต่างๆ ที่ติดตั้งไว้ก่อนหน้านี้ และผู้แจ้งจะต้องบันทึกข้อมูลชื่อ นามสกุล และเบอร์โทรศัพท์ ส่วนข้อมูลสถานที่ของอุปกรณ์ที่บันทึกไว้จะแสดงขึ้นมาโดยอัตโนมัติโดยไม่ต้องป้อนอีก ส่วนข้อมูลการชำรุดจะเป็นการระบุประเภทของอุปกรณ์ชำรุดเบื้องต้นโดยจะให้ผู้ใช้งานเลือกประเภทของอุปกรณ์แทนการพิมพ์ หรือผู้ใช้ต้องการใส่รายละเอียดเพิ่มเติมก็สามารถใส่เพิ่มได้ในช่องรายละเอียดเพิ่มเติม ( B )

#### แจ้งซ่อม

☰ ข้อมูลการแจ้งซ่อม A

หมายเลขใบแจ้งซ่อม	J180525112417		
รหัสอุปกรณ์		ชื่ออุปกรณ์	
ชื่อ		นามสกุล	
เบอร์โทรศัพท์			
อาคารที่ติดตั้ง		ชั้นที่ติดตั้ง	
ห้องที่ติดตั้ง			

☰ ข้อมูลการชำรุด B

อาคารชำรุด	<--- กรุณาเลือก --->
รายละเอียดเพิ่มเติม	

ภาพที่ 52 แสดงหน้าจองานแจ้งซ่อม

#### 4.4.10 หน้าจอมอบหมายงานซ่อม

ในหน้าจอมอบหมายงานซ่อมจะแบ่งเป็น 3 ส่วน คือส่วน ( A ) ผู้ใช้สามารถเลือกใบแจ้งซ่อมอ้างอิงได้โดยกรณีที่เป็นงานซ่อมต่อเนื่องหากไม่ใช้งานซ่อมต่อเนื่องผู้ใช้ก็ไม่ต้องระบุลงไป ผู้ใช้ระบุเวลาที่ใช้ในการซ่อมโดยมีหน่วยเป็นชั่วโมง ส่วน ( B ) จะเป็นการเลือกช่างผู้รับมอบหมายงานโดยการเลือกจากเมนูช่างผู้รับมอบหมายงานโดยระบบจะแสดงภาระงานของช่างที่เลือกขึ้นมาแสดงในตารางด้านล่างเพื่อใช้เป็นข้อมูลในการตัดสินใจมอบหมายงาน ( C ) และระบบจะทำการบันทึกข้อมูลต่างๆ

☰ มอบหมายงานการซ่อม

ใบแจ้งซ่อมอ้างอิง
<-- กรุณาเลือก --> ▼

เวลาที่ใช้ในการซ่อม (ชม.)

ช่างผู้รับมอบหมายงาน
Technician03 P-B ▼

ลักษณะของงาน
 ปกติ ด่วน

หมายเลขใบแจ้งซ่อม	รหัสอุปกรณ์	ชื่ออุปกรณ์	อาการเสีย	รายละเอียด	สถานะงานซ่อม	เวลาซ่อม(ชม.)	วันที่แจ้งซ่อม
P18052306284198	A02-No10	Split Type Unit	แผนบำรุงรักษา	ตรวจเช็คสภาพของคอมเพรสเซอร์	กำลังดำเนินการ	0.08	28/05/2018
P18052306284191	A02-No10	Split Type Unit	แผนบำรุงรักษา	ทำความสะอาดถาดเดรน และการอุดตันของท่อเดรน	กำลังดำเนินการ	0.17	28/05/2018
P18052306284186	A02-No10	Split Type Unit	แผนบำรุงรักษา	ทำความสะอาดคอยล์ร้อน , เย็น , พัดลมและตะแกรง	กำลังดำเนินการ	0.50	28/05/2018
P18052306284183	A02-No10	Split Type Unit	แผนบำรุงรักษา	ตรวจสอบเสียงและการสั่นสะเทือนของอุปกรณ์ขณะทำงาน	กำลังดำเนินการ	0.08	28/05/2018
P18052306284165	A02-No9	Split Type Unit	แผนบำรุงรักษา	ตรวจสอบเสียงและการสั่นสะเทือนของอุปกรณ์ขณะทำงาน	กำลังดำเนินการ	0.08	28/05/2018
P18052306284164	A02-No8	Split Type Unit	แผนบำรุงรักษา	ตรวจฟังก์ชันการทำงานของอุปกรณ์รีโมทควบคุม	กำลังดำเนินการ	0.08	28/05/2018
จำนวนงาน 6					รวมเวลาซ่อม(ชม.) 1		

บันทึก
กลับ
กลับมาหน้า Dashboard

ภาพที่ 53 แสดงหน้าจอมอบหมายงานซ่อม

เมื่อกดปุ่มบันทึก ( C ) จะทำการเก็บข้อมูลต่างๆ พร้อมทั้งส่งข้อมูลในการซ่อมไปยังช่างผู้ได้รับมอบหมายงาน

#### 4.4.11 หน้าจอตรวจสอบรายการแจ้งซ่อม

ในส่วนหน้าจอตรวจสอบรายการแจ้งซ่อม จะประกอบไปด้วย 3 ส่วน โดยส่วนแรกจะเป็นการใส่ข้อมูลหรือเลือกข้อมูลที่จะใช้ค้นหาไป ( A ) และเมื่อกดปุ่มค้นหา ระบบจะทำการค้นหาข้อมูลในส่วน ( B ) หลังจากนั้น โปรแกรมจะแสดงข้อมูลที่ได้จากเงื่อนไขมาแสดงในส่วนของ ( C )

##### ตรวจสอบรายการแจ้งซ่อม

ภาพที่ 54 แสดงหน้าจอตรวจสอบรายการแจ้งซ่อม

นอกจากนี้แล้วหากผู้ใช้ต้องการข้อมูลที่ ได้จากการค้นหา ก็สามารถ กดปุ่มบันทึกเป็น Excel ในส่วน ( B ) โปรแกรมจะส่งข้อมูลไปยังโปรแกรม Excel

	E	F	G	H	I	J	K	L	M
1	อาการเสีย	รายละเอียด	รายงานผลการ	สถานะงานซ่อม	ผู้ซ่อม	เวลาซ่อม	ลักษณะ	วันที่แจ้ง	แก้ไข
2	แผนบำรุงรักษา	ตรวจสอบระดับน้ำในถัง		กำลังดำเนินการ	Technician01 E-A	0	Normal	04/07/2018	
3	แผนบำรุงรักษา	ทดสอบการปิด และเปิดวาล์วต่างๆ		กำลังดำเนินการ	Technician01 E-A	0	Normal	04/07/2018	
4	งานไฟฟ้า	เปิดไม่ติด		กำลังดำเนินการ	Technician01 E-A	1	Normal	09/06/2018	
5	งาน	เสียงคังผิดปกติ		กำลังดำเนินการ	Technician01 E-A	1	Normal	24/05/2018	
6	งานประปา	น้ำรั่ว		กำลังดำเนินการ	Technician04 P-B	1	Urgent	23/05/2018	

ภาพที่ 55 แสดงข้อมูลที่ส่งไปยัง โปรแกรม Excel

#### 4.4.12 เพิ่มแผนซ่อมบำรุง(กำหนดตัวอุปกรณ์)

หน้าจอเพิ่มแผนงานบำรุงรักษาจะประกอบไปด้วย 2 ส่วน ส่วน ( A ) เป็นส่วนที่กรอกข้อมูลรหัสอุปกรณ์ลงไปเพื่อให้โปรแกรมค้นหาข้อมูลรายละเอียดต่างๆ ของอุปกรณ์ขึ้นมา โดยเมื่อผู้ใช้กรอกตัวอักษรลงไปจะทำการค้นหาให้และแสดงผลการค้นหาทางด้านล่าง [15] ในส่วน ( B ) จะเป็นการแสดงข้อมูลรายละเอียดของอุปกรณ์ที่เลือก

#### แผนงานบำรุงรักษา

☰ ค้นหาอุปกรณ์

A

ค้นหาอุปกรณ์

- A02-No1, Split Type Unit
- A02-No10, Split Type Unit
- A02-No11, Split Type Unit

ข้อมูลอุปกรณ์

B

รหัสอุปกรณ์	A02-No1	ชื่ออุปกรณ์	Split Type Unit
อาคารที่ติดตั้ง	อาคาร1	ชั้นที่ติดตั้ง	1
		ห้องที่ติดตั้ง	101

ภาพที่ 56 แสดงหน้าจอค้นหาอุปกรณ์ที่จะนำไปสร้างแผนการซ่อมบำรุง



#### 4.4.13 หน้าจอสร้างแผนซ่อมบำรุง

เมื่อทำการเลือกอุปกรณ์ในหัวข้อ 4.3.12 เสร็จแล้วต่อมาเป็นหน้าจอขั้นตอนการสร้างแผนงานบำรุงรักษาโดยแบ่งเป็นส่วนประกอบ 4 ส่วน โดยส่วนแรก (A) จะเป็นการกำหนดวันที่เริ่มใช้งานอุปกรณ์ จากนั้นเลือกช่างผู้รับผิดชอบในการดูแลแผนตรวจเช็คหรือซ่อมบำรุง ส่วน (B) จะเป็นการกำหนดลักษณะงานแผนบำรุงรักษาโดยกรอกความถี่ในการทำงานโดยมีหน่วยเป็นวันและระยะเวลาในการตรวจเช็คจากนั้นกดปุ่มเพิ่มรายละเอียดจะทำให้ข้อมูลถูกส่งไปแสดงในส่วน (C) ผู้ใช้งานสามารถทำการเพิ่มรายละเอียดและขั้นตอนการบำรุงรักษาได้เพิ่มเติม จากนั้นทำการบันทึกโดยกดปุ่มบันทึกที่ส่วน (D) เป็นการเสร็จสิ้นการสร้างแผนงานบำรุงรักษา

สร้างแผนงานบำรุงรักษา

วันที่เริ่มใช้งานอุปกรณ์  A

ช่างผู้รับผิดชอบ  B

รายละเอียดและขั้นตอนการบำรุงรักษา + เพิ่มรายละเอียด

รายละเอียดและขั้นตอนการบำรุงรักษา  ความถี่ (วัน)  เวลาในการตรวจสอบ (นาที)

รายละเอียด	ความถี่ (วัน)	เวลาในการตรวจสอบ	ลบ
ตรวจสอบเสียงและการสั่นสะเทือนของอุปกรณ์ขณะทำงาน	90	5	<span style="color: red;">✖</span>
ทำความสะอาดแผงกรองอากาศ (Air Filter) / ตัวเครื่อง	90	5	<span style="color: red;">✖</span>

C

ภาพที่ 57 แสดงหน้าจอสร้างแผนซ่อมบำรุง

#### 4.4.14 หน้าจอตรวจสอบแผนการบำรุงรักษา

หน้าจอตรวจสอบแผนการบำรุงรักษาประกอบไปด้วย 3 ส่วน โดยส่วนแรก (A) จะเป็นการกรอกข้อมูลเพื่อดูรายละเอียดของแผนงานที่ได้ทำการสร้างไว้ในหัวข้อ 4.3.13 เมื่อกรอกข้อมูลให้ทำการกดปุ่มค้นหา (B) โปรแกรมจะทำการแสดงแผนการบำรุงรักษาออกมาในส่วน (C) หากต้องการแก้ไขหรือลบแผนที่สร้างมาก็สามารถทำได้จากด้านขวาตาราง เมื่อต้องการนำข้อมูลออกมาให้กดปุ่มบันทึกเป็น Excel ในส่วน (B) โปรแกรมจะส่งข้อมูลออกไปยังโปรแกรม Excel ตรวจสอบแผนการบำรุงรักษา

ชื่ออุปกรณ์	รายละเอียดและขั้นตอน	ช่างผู้รับผิดชอบ	ความถี่ในการดูแล (วัน)	วันที่ถัดไปในการดูแล	แก้ไข	ลบ
รหัสอุปกรณ์: A02-No1						
รหัสอุปกรณ์: A02-No10						
รหัสอุปกรณ์: A02-No11						
Split Type Unit	ตรวจเช็คจำนวนที่ท่อหุ้มท่อ และข้อต่อต่างๆ ของท่อน้ำยา	Technician01 E-A	365	20/06/2019		
Split Type Unit	ตรวจเช็คและหล่อลื่น มอเตอร์และพัดลมทั้ง Fan Coil และ Condensing Unit	Technician01 E-A	180	17/12/2018		
Split Type Unit	ตรวจสอบเสียงและการสั่นสะเทือนของอุปกรณ์ขณะทำงาน	Technician01 E-A	90	18/09/2018		
Split Type Unit	ทำความสะอาดแผงกรองอากาศ (Air Filter) / ตัวเครื่อง	Technician01 E-A	30	20/07/2018		

ภาพที่ 58 แสดงหน้าจอตรวจสอบแผนการบำรุงรักษา

A	B	C	D	E	F	G	H
	ชื่ออุปกรณ์	รายละเอียดและขั้นตอน	ช่างผู้รับผิดชอบ	ความถี่ในการดูแล (วัน)	วันที่ถัดไปในการดูแล	แก้ไข	ลบ
2	รหัสอุปกรณ์: A02-No1						
3	Split Type Unit	ตรวจเช็คสภาพของคอมเพรสเซอร์	Technician02 E-A	365	30/05/2019		
4	Split Type Unit	ทำความสะอาดคอยล์ร้อน , เย็น ด้วยน้ำยาเคมี	Technician02 E-A	365	30/05/2019		
5	Split Type Unit	ตรวจเช็คจำนวนที่ท่อหุ้มท่อ และข้อต่อต่างๆ ของท่อน้ำยา	Technician02 E-A	365	30/05/2019		
6	Split Type Unit	ตรวจสอบสายไฟและข้อต่อไฟฟ้าต่างๆ ว่าหลวมหรือไม่	Technician02 E-A	365	30/05/2019		
7	Split Type Unit	ทำความสะอาดครน และการอุดตันของท่อแควน	Technician02 E-A	180	26/11/2018		
8	Split Type Unit	ตรวจเช็คและหล่อลื่น มอเตอร์และพัดลมทั้ง Fan Coil และ Condensing	Technician02 E-A	180	26/11/2018		
9	Split Type Unit	ทำความสะอาดคอยล์ร้อน , เย็น , พัดลมและตะแกรง	Technician02 E-A	180	26/11/2018		
10	Split Type Unit	ทำความสะอาดแผงกรองอากาศ (Air Filter) / ตัวเครื่อง	Technician02 E-A	90	28/08/2018		
11	Split Type Unit	ตรวจสอบเสียงและการสั่นสะเทือนของอุปกรณ์ขณะทำงาน	Technician02 E-A	90	28/08/2018		

ภาพที่ 59 แสดงข้อมูลที่ส่งไปยังโปรแกรม Excel



#### 4.4.15 หน้าจอแสดงงานที่ได้รับมอบหมายของช่าง

ซึ่งจะประกอบไปด้วยส่วนการแสดงผลข้อมูลและส่วนการรายงานผลการซ่อม โดยส่วนแรก (A) จะเป็นส่วนบอกว่า Username ของช่างที่เข้ามาในระบบ ส่วน (B) จะเป็นส่วนเลือกเงื่อนไขในการค้นหาและแสดงงานซ่อมที่ส่งมาให้ช่างดำเนินการ ส่วน (C) จะเป็นส่วนแสดงรายการงานซ่อมต่างๆที่ได้รับเข้ามา โดยช่างสามารถดูรายงานผลการซ่อมได้ในหัวข้อแก้ไข

##### ระบบบริหารและจัดการบำรุงรักษาอาคาร

หมายเลขใบแจ้งซ่อม	หมายเลขอ้างอิง	รหัสอุปกรณ์	ชื่ออุปกรณ์	อาการเสียบ	รายละเอียด	รายงานผลการซ่อม	สถานะงานซ่อม	ลักษณะงาน	วันที่แจ้งซ่อม	แก้ไข
J180518215345		A02-No9	Split Type Unit	งานเครื่องปรับอากาศ	เสียงดังผิดปกติ	ยัดน๊อตใหม่	เสร็จเรียบร้อย	Urgent	18/05/2018	
J180523183705	C	A02-No3	Split Type Unit	งานเครื่องปรับอากาศ	เปิดไม่ติด		เสร็จเรียบร้อย	Urgent	23/05/2018	

ภาพที่ 60 แสดงหน้าจองานที่ได้รับมอบหมายของช่าง

#### 4.4.16 หน้าจอผู้แจ้งซ่อม

หน้าจอของผู้แจ้งซ่อมจะประกอบไปด้วย 3 ส่วน ส่วนแรก(A) จะเป็นการแสดงผลข้อมูล Username ของผู้ใช้งาน ส่วน (B) จะเป็นส่วนแสดงงานแจ้งซ่อมของผู้ใช้งานที่ได้แจ้งไป โดยสามารถดูรายละเอียดและสถานะของงานแจ้งซ่อมได้ ส่วน (C) จะเป็นส่วนการแจ้งซ่อม

##### ระบบบริหารและจัดการบำรุงรักษาอาคาร

หมายเลขใบแจ้งซ่อม	รหัสอุปกรณ์	ชื่ออุปกรณ์	อาการเสียบ	รายละเอียด	รายงานผลการซ่อม	สถานะงานซ่อม	วันที่แจ้งซ่อม	วันที่แก้ไขล่าสุด	แก้ไข
J180609164643	faucet-001	faucet	งานประปา	มีน้ำไม่ไหล		รับแจ้งซ่อม	09/06/2018		
J180609164310	Door-001	Door	งานอาคาร	เปิดไม่ไ้		รับแจ้งซ่อม	09/06/2018		
J180518210534	A02-No2	Split Type Unit	งานเครื่องปรับอากาศ	ไม่ทำงาน	รออะไหล่	ไม่สามารถแก้ไขได้	20/05/2018	20/05/2018	
J180518205958	FL18-002	Fluorescent 18W	งานไฟฟ้า	เปิดไม่ติด	เปลี่ยนหลอด	เสร็จเรียบร้อย	20/05/2018	20/05/2018	

ภาพที่ 61 แสดงหน้าจอหลักของผู้แจ้งซ่อม

หน้าจอส่วนการแจ้งซ่อมที่ผู้แจ้งสามารถกรอกข้อมูลเองได้ โดยจะแบ่งเป็น 2 ส่วนคือ (A) ส่วนการป้อนข้อมูลต่างๆของอุปกรณ์ โดยโปรแกรมจะแสดงข้อมูลรายละเอียดของผู้แจ้งขึ้นมาให้ผู้ใช้ต้องใช้รหัสอุปกรณ์ในการแจ้งซ่อม ซึ่งสามารถตรวจสอบได้จากตัวอุปกรณ์ เมื่อกรอกข้อมูลลงไป โปรแกรมจะดึงข้อมูลสถานที่ติดตั้งของอุปกรณ์ขึ้นมาให้ จากนั้นในส่วน (B) ผู้แจ้งทำการเลือกประเภทของอุปกรณ์ที่ชำรุดและสามารถแจ้งรายละเอียดเพิ่มเติมได้

### แจ้งซ่อม

☰ ข้อมูลการแจ้งซ่อม
คุณ USRE01 To Fix

A

หมายเลขใบแจ้งซ่อม	<input type="text" value="J180525130340"/>		
รหัสอุปกรณ์	<input type="text"/>	ชื่ออุปกรณ์	<input type="text"/>
ชื่อ	<input type="text" value="USRE01"/>	นามสกุล	<input type="text" value="To Fix"/>
อาคารที่ติดตั้ง	<input type="text"/>	ชั้นที่ติดตั้ง	<input type="text"/>
		เบอร์โทรศัพท์	<input type="text"/>
		ห้องที่ติดตั้ง	<input type="text"/>

ข้อมูลการชำรุด

อาคารชำรุด	<input type="text" value="&lt;--- กรุณาเลือก ---&gt;"/>
รายละเอียดเพิ่มเติม	<input style="height: 20px;" type="text"/>

B

บันทึก
กลับหน้า Dashboard

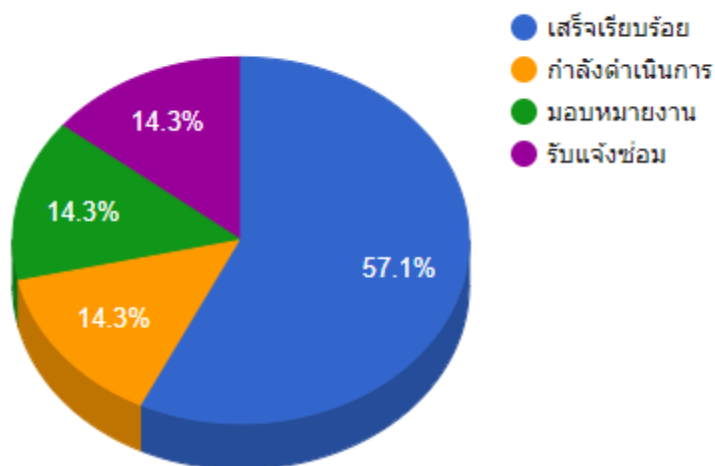
ภาพที่ 62 หน้าต่างแจ้งซ่อมสำหรับผู้แจ้งซ่อม

#### 4.4.17 หน้าจอรายงานรวม

รายงานในโปรแกรมจะมีทั้งหมด 6 แบบ ตามรูปแบบข้อมูลที่ต้องการแสดง

รายงานการซ่อมประจำวัน

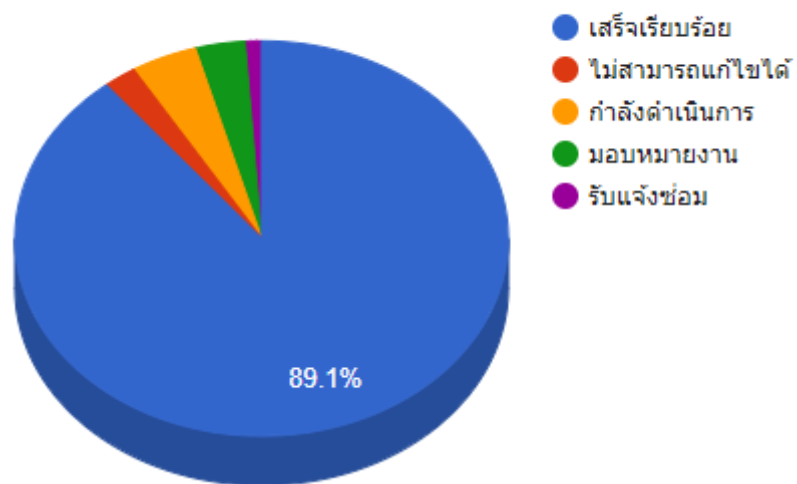
##### งานแจ้งซ่อมประจำวัน



ภาพที่ 63 รายงานการซ่อมประจำวัน

รายงานการซ่อมประจำเดือน

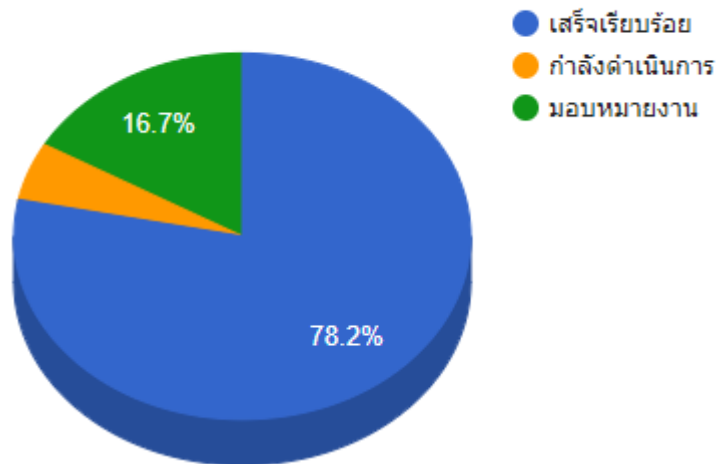
##### งานแจ้งซ่อมประจำเดือน



ภาพที่ 64 รายงานการซ่อมประจำเดือน

## รายงานแผนบำรุงรักษาประจำวัน

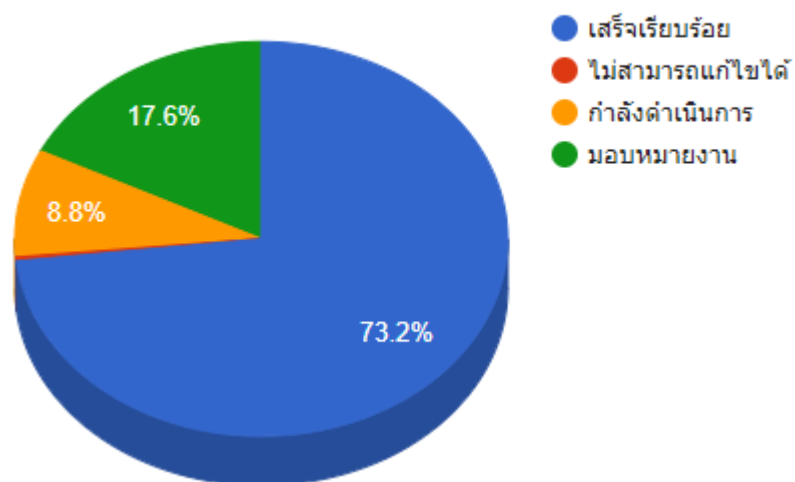
### งานแผนการดูแลรักษาวันนี้



ภาพที่ 65 รายงานแผนบำรุงรักษาประจำวัน

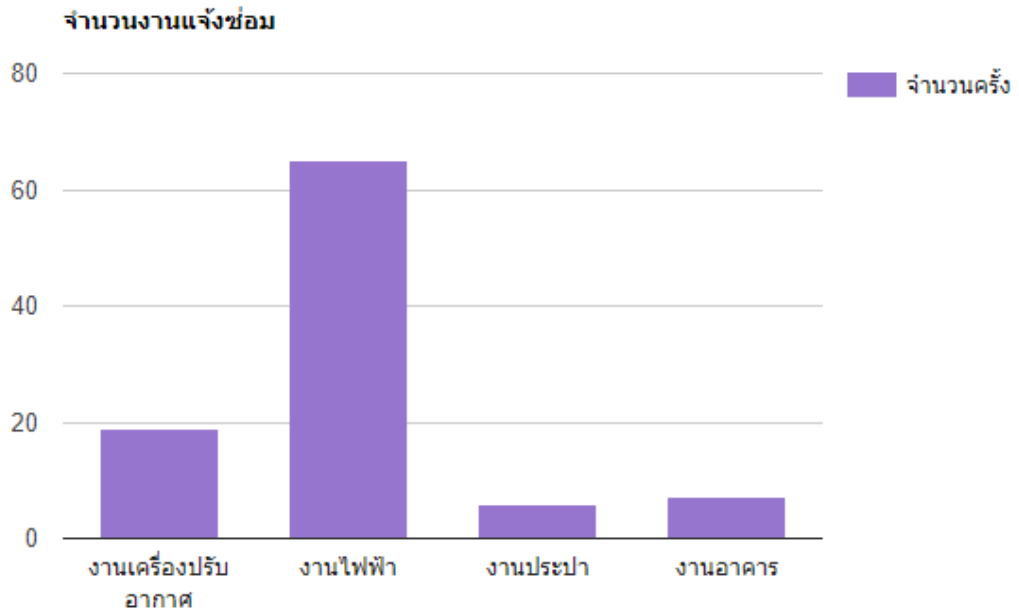
## รายงานแผนบำรุงรักษาประจำเดือน

### งานแผนการดูแลรักษาประจำเดือน



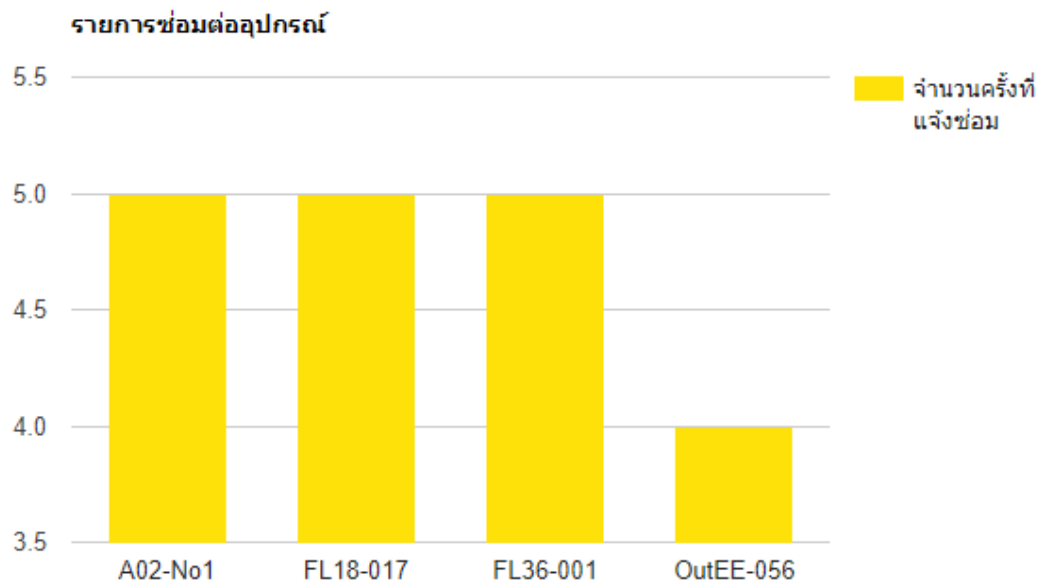
ภาพที่ 66 รายงานแผนบำรุงรักษาประจำเดือน

### รายงานจำนวนการแจ้งซ่อมแยกตามประเภทอุปกรณ์



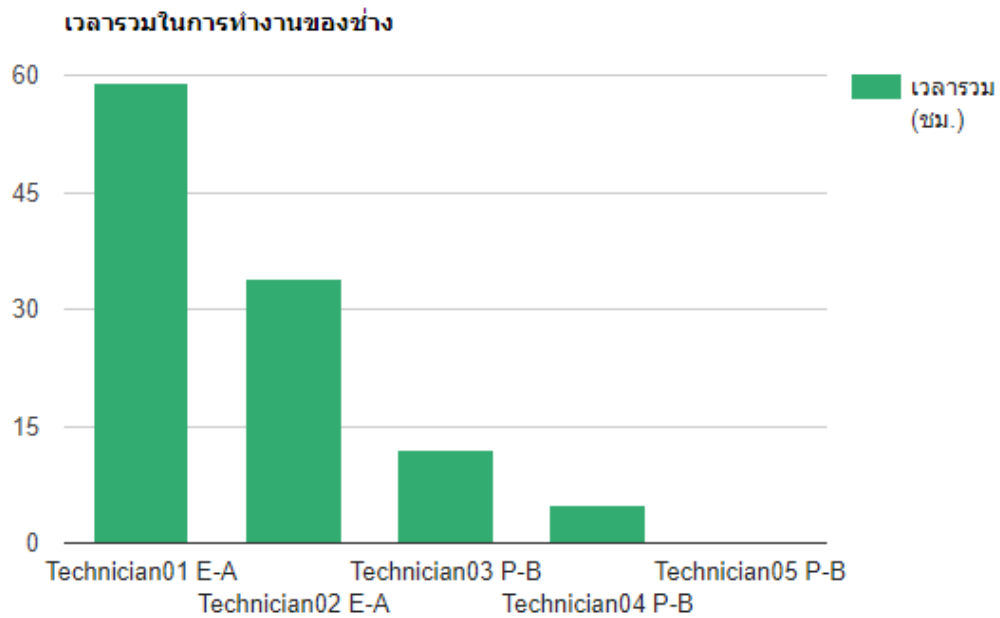
ภาพที่ 67 รายงานจำนวนการแจ้งซ่อมแยกตามประเภทอุปกรณ์

### รายงานจำนวนการซ่อมต่ออุปกรณ์



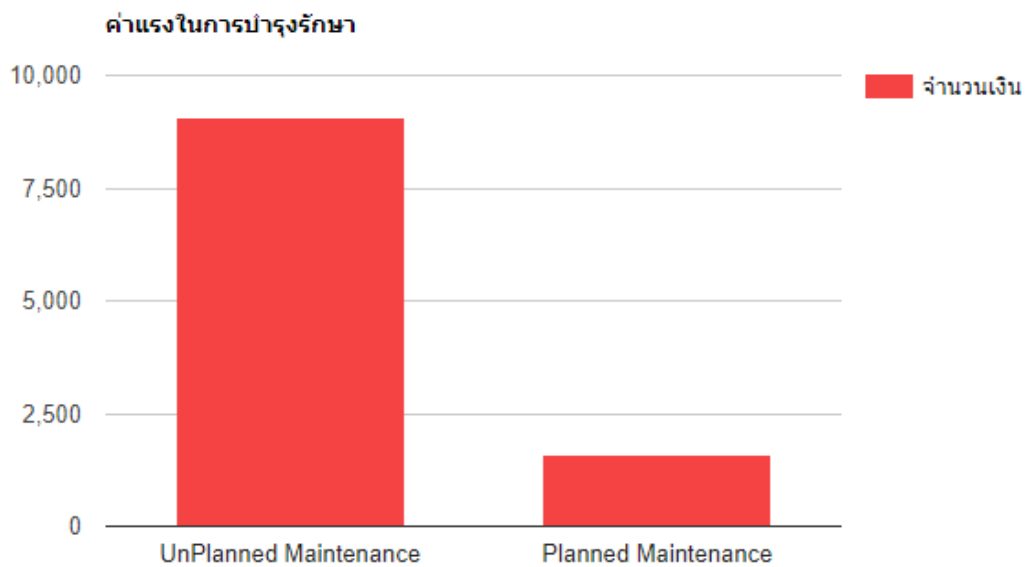
ภาพที่ 68 รายงานจำนวนการซ่อมต่ออุปกรณ์

### รายงานเวลาในการทำงานของช่าง



ภาพที่ 69 รายงานเวลาในการทำงานของช่าง

### รายงานค่าใช้จ่ายในการบำรุงรักษา



ภาพที่ 70 รายงานค่าใช้จ่ายในการบำรุงรักษา

## บทที่ 5

### สรุปผลการศึกษาและข้อเสนอแนะ

#### 5.1 สรุปผลการพัฒนาโปรแกรมซ่อมบำรุงรักษาอาคาร

จากการศึกษาระบบงานซ่อมบำรุงของอาคารสำนักงาน จะพบว่าการทำงานจะอยู่ในรูปแบบเอกสารทั้งหมดอาจมีการใช้โปรแกรมพื้นฐานต่างๆ เช่น Microsoft Excel ช่วยในการเก็บข้อมูลบ้างแต่ระบบโดยรวมยังทำงาน โดยการประสานงานทางเอกสารและการติดต่อสื่อสารทางโทรศัพท์ ซึ่งกระบวนการดังกล่าวสามารถเกิดข้อผิดพลาดได้ในหลายส่วนขั้นตอน ไม่ว่าจะเป็นการสื่อสารผิดพลาดทำให้ข้อมูลไม่ถูกต้อง การสูญหายของเอกสารความล่าช้าในระบบนำส่งแบบสารบัญ ซึ่งผลการผิดพลาดหรือล่าช้าดังกล่าวอาจส่งผลกระทบต่อในรูปแบบของการใช้เวลาการทำงานเพิ่มขึ้น หรือแม้กระทั่งการเกิดค่าใช้จ่ายที่มากขึ้นในการบำรุงรักษา นอกจากนี้แล้วระบบเอกสารยังมีข้อจำกัดในการเข้าถึงซึ่งต้องใช้เวลาในการค้นหาและรวบรวม ทำให้ผู้ดูแลหรือผู้บริการขาดข้อมูลที่จะนำมาใช้ในการวางแผนงานในอนาคตต่อไป

จากที่กล่าวมาผู้ศึกษาจึงมีแนวความคิดที่จะพัฒนาโปรแกรมคอมพิวเตอร์เพื่อนำมาใช้ในการแก้ปัญหาดังกล่าว โดยเลือกใช้เทคโนโลยีการพัฒนาโปรแกรมบนเว็บไซต์และจัดเก็บข้อมูลในฐานข้อมูลกลางเพื่อแก้ไขปัญหาในการเข้าถึงข้อมูลจากหลายบุคคล และการทำงานบนเว็บไซต์ยังช่วยให้การเข้าถึงโปรแกรมและข้อมูลได้จากทุกที่มีอินเทอร์เน็ต ซึ่งสอดคล้องกับการทำงานที่ไม่ได้ประจำอยู่กับโต๊ะทำงาน

##### 5.1.1 สรุปความสามารถของโปรแกรม ความสามารถของโปรแกรมมีรายละเอียดดังต่อไปนี้

- โปรแกรมสามารถสร้างใบแจ้งซ่อมและมอบหมายงานซ่อมได้ในระบบ
- โปรแกรมสามารถแสดงสถานะของงานแจ้งซ่อมได้ในทุกขั้นตอนตั้งแต่เริ่มต้นจนจบงาน
- โปรแกรมสามารถเพิ่มหรือลดรายการข้อมูลอุปกรณ์ประจำอาคารได้
- โปรแกรมสามารถเพิ่มหรือลด แผนการบำรุงรักษาให้กับอุปกรณ์ได้
- โปรแกรมสามารถแจ้งเตือนล่วงหน้าสำหรับงานแผนบำรุงรักษาที่ได้กำหนดไว้ตามแผน



## 5.2 ปัญหาและอุปสรรค

ในการเก็บข้อมูลจากฝ่ายซ่อมบำรุงอาคารที่ได้ดำเนินการสัมภาษณ์ในแต่ละอาคารนั้น จะพบว่าในแต่ละอาคารจะมีความแตกต่างกันในการทำงาน ไม่ว่าจะเป็นขั้นตอนการทำงาน รูปแบบการเก็บข้อมูล และนอกจากนั้นข้อมูลบางอย่างก็มีความสำคัญกับบริษัท ไม่สามารถเปิดเผยได้ซึ่งเป็นอุปสรรคสำคัญในการออกแบบฐานข้อมูลที่ใช้ในโปรแกรม ทำให้เกิดความไม่สมบูรณ์และไม่ครอบคลุมของฐานข้อมูลในบางเรื่อง ซึ่งเป็นข้อจำกัดในการทำวิทยานิพนธ์ฉบับนี้

## 5.3 ข้อเสนอแนะ

### 5.3.1 ข้อเสนอแนะจากกลุ่มผู้ใช้งานโปรแกรมแบ่งเป็น 3 กลุ่ม

#### 5.3.1.1 กลุ่มหัวหน้าช่างหรือเจ้าหน้าที่ผู้ทำหน้าที่รับแจ้งงาน (Admin)

เป็นโปรแกรมที่สามารถช่วยด้านการจัดการงานซ่อมบำรุงได้สะดวกขึ้น และช่วยในเรื่องการจัดเก็บข้อมูลทำรายงานเสนอผู้บริหารได้อย่างรวดเร็ว

#### 5.3.1.2 กลุ่มช่างที่ได้รับมอบหมายงานซ่อม (Technician)

การมอบหมายงานผ่านระบบทำให้ช่างสามารถเห็นงานที่ได้รับมอบหมายได้ทันทีและรายงานผลการซ่อมได้จากหน้าจอโดยไม่ต้องเดินทางกลับมาส่งใบแจ้งซ่อมและรับใบแจ้งซ่อมใหม่ที่สำนักงาน

#### 5.3.1.3 กลุ่มผู้แจ้งซ่อม (User)

ช่วยอำนวยความสะดวกในการแจ้งซ่อมได้มากขึ้นจากกระบวนการแจ้งซ่อมด้วยตนเองในระบบเพื่อแก้ไขปัญหาการติดต่อกับฝ่ายแจ้งซ่อมไม่ได้

5.3.2 พัฒนาโปรแกรมในส่วนการแสดงผล (User Interface) โดยลดการเปิดหน้าต่างใหม่ให้น้อยลง ทำให้โปรแกรมไม่ซับซ้อนและใช้งานง่ายขึ้น

5.3.3 พัฒนาโปรแกรมให้สามารถเพิ่มหรือกำหนดตัวเลือกในการสร้างรายงานได้โดยผู้ดูแลระบบ

5.3.4 พัฒนาโปรแกรมให้สามารถนำเข้าข้อมูลในรูปแบบไฟล์ข้อมูลได้โดยตรงในลักษณะ CSV ไฟล์

5.3.5 พัฒนาระบบส่วนการแจ้งซ่อมของ ผู้แจ้งและช่างในรูปแบบแอปพลิเคชันบนสมาร์ตโฟนเพื่ออำนวยความสะดวกและใช้งานได้ง่ายขึ้น



## 1. ชุดคำสั่งหน้าจอลงชื่อ Login.aspx

**ระบบบริหารและจัดการบำรุงรักษาอาคาร**

USERNAME  ← บ้อนชื่อผู้ใช้ (1)

PASSWORD  ← บ้อนรหัสผ่าน (2)

เข้าใช้งาน →   ← ลงทะเบียน

ภาพที่ 71 หน้าจอ Login

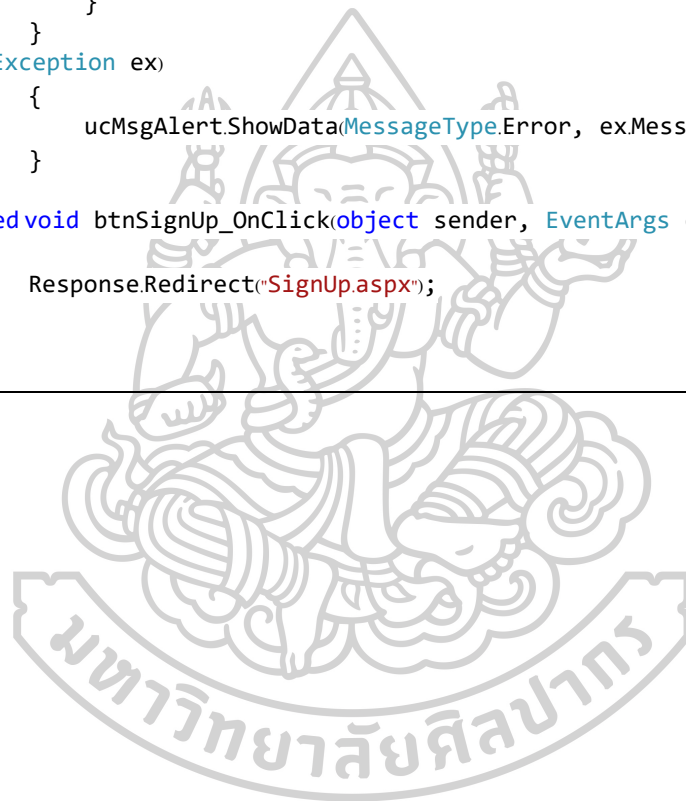
Login.aspx.cs

```
namespace MaintenanceSystem
{
    public partial class Login : System.Web.UI.Page
    {
        protected void btnLogin_OnClick(object sender, EventArgs e)
        {
            try
            {
                var service = new Database();
                var result = service.GetUserLogin(txtUsername.Text.Trim(),
                txtPassword.Text.Trim());
                if (!result)
                    throw new Exception("กรุณาตรวจสอบ USERNAME PASSWORD");

                var usertype = service.CheckTypeUserName(txtUsername.Text.Trim());
                if (usertype == UserType.Backoffice)
                {
                    var obj = service.QueryUserInfo(txtUsername.Text.Trim());
                    if (obj.Rows.Count > 0)
                    {
                        Session["Username"] = obj.Rows[0]["Username"];
                        Session["Name"] = obj.Rows[0]["Name"];
                        Session["LastName"] = obj.Rows[0]["LastName"];
                    }

                    Response.Redirect(ResolveClientUrl("DashBoard.aspx"));
                }
                else if (usertype == UserType.Technician)
                {
                    var obj =
                    service.GetrowTechnicianProfileByUsername(txtUsername.Text.Trim());
                    if (obj.Rows.Count > 0)
                    {
                        Session["Username"] = obj.Rows[0]["Username"];
                        Session["Name"] = obj.Rows[0]["FirstName"];
                        Session["LastName"] = obj.Rows[0]["LastName"];
                    }
                }
            }
        }
    }
}
```

```
        Response.Redirect("DashBoardTechnician.aspx");
    }
else
    {
var obj = service.QueryUserInfo(txtUsername.Text.Trim());
    if(obj.Rows.Count > 0)
        {
            Session["Username"] = obj.Rows[0]["Username"];
            Session["Name"] = obj.Rows[0]["Name"];
            Session["LastName"] = obj.Rows[0]["LastName"];
        }
        Response.Redirect("DashBoardUser.aspx");
    }
}
catch(Exception ex)
    {
        ucMsgAlert.ShowData(MessageType.Error, ex.Message);
    }
}
protected void btnSignUp_OnClick(object sender, EventArgs e)
    {
        Response.Redirect("SignUp.aspx");
    }
}
}
```



## 2. ชุดคำสั่งหน้า Dashboard.aspx.cs

ยินดีต้อนรับ คุณ สมใจ สุขสม ออกจากระบบ

รายละเอียดงานแจ้งซ่อม วันนี้

ลักษณะของงาน  ทั้งหมด  ปกติ  ด่วน  
 อาการชำรุด  ทั้งหมด  งานไฟฟ้า  งานเครื่องปรับอากาศ  งานประปา  งานอาคาร  งานอื่นๆ  แผนบำรุงรักษา  
 สถานะงานซ่อม  ทั้งหมด  แจ้งซ่อม  มอบหมายงานให้ช่าง  ช่างกำลังตรวจสอบงานแจ้งซ่อม  แก้ไขเรียบร้อยแล้ว  ไม่สามารถแก้ไขได้

หมายเลขใบแจ้งซ่อม	หมายเลขอ้างอิง	ชื่ออุปกรณ์	อาการเสีย	รายละเอียด	รายงานผลการซ่อม	สถานะงานซ่อม	ลักษณะงาน	ผู้ซ่อม	เวลาซ่อม(ชม.)	วันที่แจ้งซ่อม	แก้ไข
J180514154904		Fluorescent 18W	งานไฟฟ้า	เปิดไม่ติด		มอบหมายงานให้ช่าง	Normal	tech01 tech01	1	14/05/2018	
J180514155023		Fluorescent 18W	งานไฟฟ้า	เปิดไม่ติด		มอบหมายงานให้ช่าง	Normal	tech01 tech01	1	14/05/2018	

รายละเอียดแผนบำรุงรักษา ล่วงหน้า 7 วัน สร้างใบแจ้งซ่อมสำหรับแผนการบำรุงรักษาวันนี้

รหัสอุปกรณ์	ชื่ออุปกรณ์	รายละเอียดและขั้นตอน	ช่างผู้รับผิดชอบ	วันที่ในการดูแล	แก้ไข
CCTV-001	CCTV	ตรวจสอบภาพ	ประเสริฐ ชัยชนะ ปรายน	29/04/2018	

ภาพที่ 72 หน้าจอ Dashboard

```
namespace MaintenanceSystem
{
    public partial class Dashboard : System.Web.UI.Page
    {
        public DataTable DashboardMaintenanceJobDatasource
        {
            get
            {
                if(Session["DashboardMaintenanceJobDatasource"] == null)
                    Session["DashboardMaintenanceJobDatasource"] = new DataTable();
                return (DataTable)Session["DashboardMaintenanceJobDatasource"];
            }
            set { Session["DashboardMaintenanceJobDatasource"] = value; }
        }
        public DataTable DashboardPlanMaintenanceDatasource
        {
            get
            {
                if(Session["DashboardPlanMaintenanceDatasource"] == null)
                    Session["DashboardPlanMaintenanceDatasource"] = new
                    DataTable();
                return (DataTable)Session["DashboardPlanMaintenanceDatasource"];
            }
            set { Session["DashboardPlanMaintenanceDatasource"] = value; }
        }
        protected void Page_Load(object sender, EventArgs e)
        {
            if(!IsPostBack)
            {
                if(Session["Name"] != null && Session["LastName"] != null)
                    lblName.Text = Session["Name"] + " " + Session["LastName"];
            }
        }
    }
}
```

```

else Response.Redirect("Login.aspx");

        DashboardMaintenanceJobDatasource = null;
        DashboardPlanMaintenanceDatasource = null;
var service = new Database();
var dtstatus = service.QueryMaintenanceStatus();
if(dtstatus.Rows.Count > 0)
    {
        cbxStatus.DataSource = dtstatus;
        cbxStatus.DataValueField = "Id";
        cbxStatus.DataTextField = "Status";
        cbxStatus.DataBind();

        for(int i =0; i < cbxStatus.Items.Count; i++)
            {
                cbxStatus.Items[i].Selected = true;
            }
    }
var dtDefected = service.QueryDefectedStuff();
if(dtDefected.Rows.Count > 0)
    {
        cbxDefect.DataSource = dtDefected;
        cbxDefect.DataValueField = "Id";
        cbxDefect.DataTextField = "Detail";
        cbxDefect.DataBind();

        for(int i =0; i < cbxDefect.Items.Count; i++)
            {
                cbxDefect.Items[i].Selected = true;
            }
    }
    ShowDataMaintenanceJob();
    ShowDataPlanMaintenance();
else
    {
        BindGridPlanMaintenance();
        BindGridMaintenanceJob();
    }
}
public void ShowDataPlanMaintenance()
{
var service = new Database();
var dt = service.QueryPlanMaintenanceBeforeToday();
    DashboardPlanMaintenanceDatasource = dt;
    BindGridPlanMaintenance();
}
public void BindGridPlanMaintenance()
{
    grdPlan.DataSource = DashboardPlanMaintenanceDatasource;
    grdPlan.DataBind();
}
}

```

```

public void ShowDataMaintenanceJob()
{
    var service = new Database();
    var status = "";
    for (int i = 0; i < cbxStatus.Items.Count; i++)
    {
        if (cbxStatus.Items[i].Selected && cbxStatus.Items[i].Value != "")
        {
            status += status == string.Empty ? cbxStatus.Items[i].Value : ","
+ cbxStatus.Items[i].Value;
        }
    }
    var jobtype = "";
    if (cbxJobtypeepAll.Checked)
    {
        if (cbxJobTypeNormal.Checked)
            jobtype = "Normal,";
        if (cbxJobTypeUrgent.Checked)
            jobtype += "Urgent";
    }
    var defected = "";
    for (int i = 0; i < cbxDefect.Items.Count; i++)
    {
        if (cbxDefect.Items[i].Selected && cbxDefect.Items[i].Value != "")
        {
            defected += defected == string.Empty ? cbxDefect.Items[i].Value
: "," + cbxDefect.Items[i].Value;
        }
    }
    var dt = service.QueryMaintenanceJobToday(jobtype, status, defected);
    DashboardMaintenanceJobDatasource = dt;
    BindGridMaintenanceJob();
}
public void BindGridMaintenanceJob()
{
    grdMaintenanceJob.DataSource = DashboardMaintenanceJobDatasource;
    grdMaintenanceJob.DataBind();
}
protected void grdMaintenanceJob_OnRowCommand(object sender,
ASPXGridViewRowCommandEventArgs e)
{
    var getid = Convert.ToInt32(e.CommandArgs.CommandArgument.ToString());
    var maintenanceCode = grdMaintenanceJob.GetRowValuesByKeyValue(getid,
"MaintenanceCode").ToString();
    Response.Redirect("MaintenanceJob.aspx?data=" + maintenanceCode);
}
protected void grdPlan_OnRowCommand(object sender,
ASPXGridViewRowCommandEventArgs e)
{
    var getid = e.CommandArgs.CommandArgument.ToString();
    var equipmentcode = grdPlan.GetRowValuesByKeyValue(getid,
"EquipmentCode").ToString();
}

```

```

        Response.Redirect("AddPlanMaintenance.aspx?data="+ equipmentcode);
    }
    protected void btnCreateToday_OnClick(object sender, EventArgs e)
    {
        var service = new Database();

        var dt = service.QueryMaintenanceMainToday();
        if(dt.Rows.Count > 0)
        {
            for(int i =0; i < dt.Rows.Count; i++)
            {
                var code = "P"+DateTime.Now.ToString("yyMMddHHmssfff");
                service.InsertMaintenanceJob(code, Session["Name"].ToString(),
                    Session["LastName"].ToString(),
                    null, Convert.ToDecimal(Convert.ToInt32(dt.Rows[i]["HoursForFixed"])/60),
                    dt.Rows[i]["Details"].ToString(),
                    dt.Rows[i]["Ref_EquipmentCode"].ToString(), 13,
                    Convert.ToInt32(dt.Rows[i]["Ref_TechnicianId"]), "",
                    Session["Username"].ToString(), Session["Username"].ToString(), JobType.Normal,
                    MaintenanceStatus.มอบหมายงานให้ช่าง,
                    Convert.ToDateTime(dt.Rows[i]["MaintenanceDate"].ToString()));
                service.UpdateMaintenancePlanDetail(DateTime.Now.AddDays(7));
                ShowDataMaintenanceJob();
                ShowDataPlanMaintenance();
                ucMsgAlert.ShowData(MessageType.Complete, "ทำการกรอกริชบร้อย");
            }
        }
        else
        {
            ucMsgAlert.ShowData(MessageType.Error, "ไม่มีรายการบำรุงรักษา");
        }
    }
    protected void cbxJob_OnSelectedIndexChanged(object sender, EventArgs e)
    {
        ShowDataMaintenanceJob();
    }
    [WebMethod]
    public static List<ReportPieStruct> ReportDailyJob()
    {
        var service = new Database();
        return service.ReportDailyJob();
    }
    [WebMethod]
    public static List<ReportPieStruct> ReportDailyPlan()
    {
        var service = new Database();
        return service.ReportDailyPlan();
    }
}

```



### 3. ชุดคำสั่งหน้า AddPlanMaintenance.aspx.cs

สร้างแผนงานบำรุงรักษา

วันที่เริ่มแผน

ช่างผู้รับผิดชอบ

รายละเอียดและขั้นตอนการบำรุงรักษา [+ เพิ่มรายละเอียด](#)

รายละเอียดและขั้นตอนการบำรุงรักษา

ความถี่ (วัน)

เวลาในการตรวจสอบ

รายละเอียด	ความถี่ (วัน)	เวลาในการตรวจสอบ	ลบ
เช็คเสียง	90	2	

[บันทึก](#) [กลับ](#) [กลับหน้า Dashboard](#)

ภาพที่ 73 หน้าจอ AddPlanMaintenance

```
namespace MaintenanceSystem
{
    public partial class AddPlanMaintenance : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                var service = new Database();
                var dtPosition = service.QueryTechnicianProfile();
                if (dtPosition.Rows.Count > 0)
                {
                    DataRow row = dtPosition.NewRow();
                    row["Id"] = "0";
                    row["PostionDescription"] = "<--- กรุณาเลือก --->";
                    dtPosition.Rows.InsertAt(row, 0);
                    ddlPosition.DataSource = dtPosition;
                    ddlPosition.DataValueField = "Id";
                    ddlPosition.DataTextField = "PostionDescription";
                    ddlPosition.DataBind();
                }
                if (Request.QueryString["data"] != null)
                {
                    var dt =
                    service.GetrowPlanMaintenanceByEquipmentCode(Request.QueryString["data"]);
                    if (dt.Rows.Count > 0)
                    {
                        txtEquipmentCode.Text =
                        dt.Rows[0]["EquipmentCode"].ToString();
                        txtEquipmentName.Text =
                        dt.Rows[0]["EquipmentName"].ToString();
                        txtBuilding.Text = dt.Rows[0]["Building"].ToString();
                        txtFloor.Text = dt.Rows[0]["Floor"].ToString();
                    }
                }
            }
        }
    }
}
```

```

        txtRoom.Text = dt.Rows[0]["Room"].ToString();
        ddlPosition.SelectedValue =
dt.Rows[0]["Ref_TechnicianId"].ToString();
        txtDatePicker.Text =
Convert.ToDateTime(dt.Rows[0]["MaintenanceDate"].ToString()).ToString("dd/MM/yyyy");
    }
    else
    {
        var dtequipment =
service.GetrowEquipmentDetails(Request.QueryString("data"));
        if(dtequipment.Rows.Count > 0)
        {
            txtEquipmentCode.Text =
dtequipment.Rows[0]["EquipmentCode"].ToString();
            txtEquipmentName.Text =
dtequipment.Rows[0]["Name"].ToString();
            txtBuilding.Text =
dtequipment.Rows[0]["Building"].ToString();
            txtFloor.Text = dtequipment.Rows[0]["Floor"].ToString();
            txtRoom.Text = dtequipment.Rows[0]["Room"].ToString();
        }
    }
}
}
}

[WebMethod]
public static List<EquipmentStruct> GetEquipmentDetails(string code)
{
    var service = new Database();
    return service.QuerySearchEquipmentDetails(code);
}

[WebMethod]
public static List<PlanMaintenanceDetailStruct>
QueryPlanMaintenanceDetails(string code)
{
    var service = new Database();
    return service.ListGetrowPlanMaintenanceByEquipmentCode(code);
}

[WebMethod]
public static void Save(List<_PlanMaintenanceDetailStruct> param, int
position, string equipmentcode, DateTime maintenancedate)
{
    var service = new Database();
    service.UpdateMaintenancePlanDetails(position, equipmentcode,
maintenancedate, param);
}
}
}
}

```

#### 4. ชุดคำสั่งหน้า ReportMaintenanceJob.aspx.cs

ตรวจสอบรายการแจ้งซ่อม

☰ ข้อมูลรายงานแจ้งซ่อม

หมายเลขใบแจ้งซ่อม  ช่างผู้รับมอบหมายงาน <--- กรุณาเลือก --->

รหัสอุปกรณ์  ชื่ออุปกรณ์

ลักษณะของงาน  ปกติ  ด่วน

สถานะงานซ่อม  ทั้งหมด  รับแจ้งซ่อม  มอบหมายงาน  กำลังดำเนินการ  เสร็จเรียบร้อย  ไม่สามารถแก้ไขได้

หมายเลขใบแจ้งซ่อม	หมายเลขอ้างอิง	รหัสอุปกรณ์	ชื่ออุปกรณ์	อาการเสีย	รายละเอียด	รายงานผลการซ่อม	สถานะงานซ่อม	ผู้ซ่อม	เวลาซ่อม(ชม.)	ลักษณะงาน	วันที่แจ้งซ่อม	แก้ไข
P18071113304802		A05-No3	Expansion Tank	แผนบำรุงรักษา	ตรวจสอบระดับน้ำในถัง		กำลังดำเนินการ	Technician01 E-A	0.00	Normal	04/07/2018	
P18071113304795		A05-No2	Expansion Tank	แผนบำรุงรักษา	ทดสอบการปิดและเปิดวาล์วต่างๆ		กำลังดำเนินการ	Technician01 E-A	0.00	Normal	04/07/2018	

ภาพที่ 74 ภาพหน้าจอตรวจสอบรายการแจ้งซ่อม

```

namespace MaintenanceSystem
{
    public partial class ReportMaintenanceJob : System.Web.UI.Page
    {
        public DataTable ReportMaintenanceJobDatasource
        {
            get
            {
                if (Session["ReportMaintenanceJobDatasource"] == null)
                    Session["ReportMaintenanceJobDatasource"] = new DataTable();
                return (DataTable)Session["ReportMaintenanceJobDatasource"];
            }
            set { Session["ReportMaintenanceJobDatasource"] = value; }
        }
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                ReportMaintenanceJobDatasource = null;
                var service = new Database();
                var dtPosition = service.QueryTechnicianProfile();
                if (dtPosition.Rows.Count > 0)
                {
                    DataRow row = dtPosition.NewRow();
                    row["Id"] = "0";
                    row["PostionDescription"] = "<--- กรุณาเลือก --->";
                    dtPosition.Rows.InsertAt(row, 0);

                    ddlPosition.DataSource = dtPosition;
                    ddlPosition.DataValueField = "Id";
                    ddlPosition.DataTextField = "PostionDescription";
                    ddlPosition.DataBind();
                }
            }
        }
    }
}

```

```

var dtstatus = service.QueryMaintenanceStatus();
if(dtstatus.Rows.Count > 0)
    {
        cbxStatus.DataSource = dtstatus;
        cbxStatus.DataValueField = "Id";
        cbxStatus.DataTextField = "Status";
        cbxStatus.DataBind();

        for(int i =0; i < cbxStatus.Items.Count; i++)
            {
                cbxStatus.Items[i].Selected = true;
            }
        }

        ShowData();
    }
else
    {
        ucMsgAlert.Hide();
        BindGrid();
    }
}

protected void btnSearch_OnClick(object sender, EventArgs e)
    {
        ShowData();
    }

protected void grd_OnRowCommand(object sender,
ASPGridViewRowCommandEventArgs e)
    {
        var getid = Convert.ToInt32(e.CommandArgs.CommandArgument.ToString());
        var maintenanceCode = grd.GetRowValuesByKeyValue(getid,
"MaintenanceCode").ToString();
        Response.Redirect("MaintenanceJob.aspx?data="+ maintenanceCode);
    }

public void ShowData()
    {
        var service = new Database();
        var status = "";
        for(int i =0; i < cbxStatus.Items.Count; i++)
            {
                if(cbxStatus.Items[i].Selected && cbxStatus.Items[i].Value != "")
                    {
                        status += status == string.Empty ? cbxStatus.Items[i].Value : ","
+ cbxStatus.Items[i].Value;
                    }
            }
        var jobtype = "";
        if(cbxJobTypeNormal.Checked)
            jobtype = "Normal,";
        if(cbxJobTypeUrgent.Checked)
            jobtype += "Urgent";
    }

```

```
var dt = service.QueryMaintenanceJob(txtMaintenanceCode.Text.Trim(),
ddlPosition.SelectedValue == "0" ? "" :
ddlPosition.SelectedValue,txtEquipmentCode.Text.Trim(),txtEquipmentName.Text.Trim()
,jobtype,status);
    ReportMaintenanceJobDatasource =dt;
    BindGrid();
}
public void BindGrid()
{
    grd.DataSource =ReportMaintenanceJobDatasource;
    grd.DataBind();
}

protected void btnExport_OnClick(object sender, EventArgs e)
{
    try
    {
        if(grd.VisibleRowCount < 1)
            throw new Exception("ไม่มีข้อมูล");
        string fileName ="ReportMaintenance";
        GridExportWriteXlsToResponse(fileName);
    }
    catch(Exception ex)
    {
        ucMsgAlert.ShowData(MessageType.Error, ex.Message);
    }
}
}
```



### 5. ชุดคำสั่งสร้าง QR\_Code อ้างอิงไฟล์ AddEquipment.aspx.cs

```

string path = Server.MapPath("~/QRCode/");
QRCodeEncoder encoder = new QRCodeEncoder();

encoder.QRCodeErrorCorrect =
QRCodeEncoder.ERROR_CORRECTION.H;
encoder.QRCodeScale = 10;

Bitmap img = encoder.Encode(txtEquipmentCode.Text);

var filename = txtEquipmentCode.Text.Trim() + ".png";
img.Save(path + filename, ImageFormat.Png);

imgqrcode.Attributes["src"] = "~/QRCode/" + filename;

```

### 6. ชุดคำสั่ง Database.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Configuration;
using System.Data;
using System.Data.Entity;
using System.Data.OleDb;
using System.Data.SqlClient;
using System.Linq;
using System.Net.NetworkInformation;
using System.Web;
using Data.OleDb;
using MaintenanceSystem.Structure;
using Microsoft.SqlServer.Server;

namespace MaintenanceSystem.Service
{
    public class Database
    {
        public Boolean GetUserLogin(string username, string password)
        {
            string strSQL = "exec CheckUserInfo ?, ?";
            using (var sqlDB = new SqlDb())
            {
                OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
                cmd.Parameters.Clear();
                cmd.Parameters.AddWithValue("", username);
                cmd.Parameters.AddWithValue("", password);
                return Convert.ToBoolean(sqlDB.GetDataRow(cmd)[0]);
            }
        }
    }
}

```

```

public Boolean ValidateDuplicateUserName(string username)
{
    string strSQL = "exec ValidateDuplicateUserName ?";
    using(var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", username);
        return Convert.ToBoolean(sqlDB.GetDataRow(cmd)(0));
    }
}

public string CheckTypeUserName(string username)
{
    string strSQL = "exec CheckTypeUserName ?";
    using(var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", username);
        return Convert.ToString(sqlDB.GetDataRow(cmd)(0));
    }
}

public DataTable QueryUserInfo(string username)
{
    string strSQL = "exec QueryUserInfo ?";
    using(var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", username);
        return sqlDB.GetDataTable(cmd);
    }
}

public DataTable GetrowTechnicianProfileByUsername(string username)
{
    string strSQL = "exec GetrowTechnicianProfileByUsername ?";
    using(var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", username);
        return sqlDB.GetDataTable(cmd);
    }
}

public DataTable QueryPosition()
{
    string strSQL = "exec QueryPosition";
    using(var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        return sqlDB.GetDataTable(cmd);
    }
}

```

```

public DataTable QueryTechnicianProfile()
{
    string strSQL = "exec QueryTechnicianProfile";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        return sqlDB.GetDataTable(cmd);
    }
}

public DataTable QueryTechnicianProfileAll(string ref_position, string name)
{
    string strSQL = "exec QueryTechnicianProfileAll ?,?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", ref_position);
        cmd.Parameters.AddWithValue("", name);
        return sqlDB.GetDataTable(cmd);
    }
}

public DataTable GetrowTechnicianProfile(int id)
{
    string strSQL = "exec GetrowTechnicianProfile ?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", id);
        return sqlDB.GetDataTable(cmd);
    }
}

public DataTable QueryEquipmentDetails(string code, string name)
{
    string strSQL = "exec QueryEquipmentDetails ?,?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", code);
        cmd.Parameters.AddWithValue("", name);
        return sqlDB.GetDataTable(cmd);
    }
}

public DataTable GetrowEquipmentDetails(string code)
{
    string strSQL = "exec GetrowEquipmentDetail ?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", code);
        return sqlDB.GetDataTable(cmd);
    }
}

```



```

    }
}
public List<EquipmentStruct> QuerySearchEquipmentDetails(string code)
{
    string strSQL = "exec [QueryEquipmentDetailforSearch]?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", code);
        DataTable dt = sqlDB.GetDataTable(cmd);
        var list = new List<EquipmentStruct>();
        list.AddRange(from DataRow dr in dt.Rows select new
EquipmentStruct(dr));
        return list;
    }
}
public DataTable QueryDefectedStuff()
{
    string strSQL = "exec QueryDefectedStuff";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        return sqlDB.GetDataTable(cmd);
    }
}
public DataTable QueryMaintenanceRef(string maintenanceCode)
{
    string strSQL = "exec [QueryMaintenanceRef]?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", maintenanceCode);
        return sqlDB.GetDataTable(cmd);
    }
}
public DataTable QueryMaintenanceJob(string code, string technicianid, string
equipmentcode, string equipmentname, string jobtype, string status)
{
    string strSQL = "exec [QueryMaintenanceJob]? ,? ,? ,? ,? ,?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", code);
        cmd.Parameters.AddWithValue("", technicianid);
        cmd.Parameters.AddWithValue("", equipmentcode);
        cmd.Parameters.AddWithValue("", equipmentname);
        cmd.Parameters.AddWithValue("", jobtype);
        cmd.Parameters.AddWithValue("", status);
        return sqlDB.GetDataTable(cmd);
    }
}

```

```

    }
    public DataTable QueryMaintenanceJob_ByTechnician(string code, string
technicianid, string equipmentcode, string equipmentname, string jobtype,
string status)
    {
        string strSQL = "exec [QueryMaintenanceJob_ByTechnician]?, ?, ?, ?, ?, ?";
        using (var sqlDB = new SqlDb())
        {
            OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", code);
            cmd.Parameters.AddWithValue("", technicianid);
            cmd.Parameters.AddWithValue("", equipmentcode);
            cmd.Parameters.AddWithValue("", equipmentname);
            cmd.Parameters.AddWithValue("", jobtype);
            cmd.Parameters.AddWithValue("", status);

            return sqlDB.GetDataTable(cmd);
        }
    }
    public DataTable QueryMaintenanceJobForUser(string user)
    {
        string strSQL = "exec [QueryMaintenanceJobForUser]?";
        using (var sqlDB = new SqlDb())
        {
            OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", user);

            return sqlDB.GetDataTable(cmd);
        }
    }
    public DataTable QueryMaintenanceJobForTechnician(string user, string
status, string type)
    {
        string strSQL = "exec [QueryMaintenanceJobForTechnician]?, ?, ?";
        using (var sqlDB = new SqlDb())
        {
            OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", user);
            cmd.Parameters.AddWithValue("", status);
            cmd.Parameters.AddWithValue("", type);

            return sqlDB.GetDataTable(cmd);
        }
    }
    public DataTable QueryMaintenanceJobToday(string jobtype, string
status, string defected)
    {
        string strSQL = "exec QueryMaintenanceJobToday ?, ?, ?";
        using (var sqlDB = new SqlDb())
        {
            OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", jobtype);

```

```

        cmd.Parameters.AddWithValue("", defected);
        cmd.Parameters.AddWithValue("", status);
    return sqlDB.GetDataTable(cmd);
    }
}
public DataTable QueryMaintenanceStatus()
{
    string strSQL = "exec QueryMaintenanceStatus";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        return sqlDB.GetDataTable(cmd);
    }
}
public DataTable GetrowMaintenanceJobByMaintenanceCode(string code)
{
    string strSQL = "exec GetrowMaintenanceJobByMaintenanceCode ?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", code);
        return sqlDB.GetDataTable(cmd);
    }
}
public DataTable QueryPlanMaintenance(string equipmentcode, string
equipmentname)
{
    string strSQL = "exec QueryPlanMaintenance ?,?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", equipmentcode);
        cmd.Parameters.AddWithValue("", equipmentname);
        return sqlDB.GetDataTable(cmd);
    }
}
public DataTable QueryPlanMaintenanceBeforeToday()
{
    string strSQL = "exec QueryPlanMaintenanceToday";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        return sqlDB.GetDataTable(cmd);
    }
}
public DataTable GetrowPlanMaintenanceById(int id)
{
    string strSQL = "exec GetrowPlanMaintenanceById ?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);

```

```

        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", id);
        return sqlDB.GetDataTable(cmd);
    }
}
public DataTable GetrowPlanMaintenanceByEquipmentCode(string equipmentcode)
{
    string strSQL = "exec GetrowPlanMaintenanceByEquipmentCode ?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", equipmentcode);
        return sqlDB.GetDataTable(cmd);
    }
}

public DataTable QueryMaintenanceMainToday()
{
    string strSQL = "exec QueryMaintenanceMainToday";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        return sqlDB.GetDataTable(cmd);
    }
}

public List<PlanMaintenanceDetailStruct>
ListGetrowPlanMaintenanceByEquipmentCode(string code)
{
    string strSQL = "exec [GetrowPlanMaintenanceByEquipmentCode] ?";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", code);
        DataTable dt = sqlDB.GetDataTable(cmd);
        var list = new List<PlanMaintenanceDetailStruct>();
        list.AddRange(from DataRow dr in dt.Rows select new
PlanMaintenanceDetailStruct(dr));
        return list;
    }
}

public void InsertTechnicianProfile(string firstname, string lastname,
string address, string tel, int reposition, string username, string
password, decimal salary)
{
    using (var sqlDb = new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            const string strSql = "exec InsertTechnicalProfile ?,?,?,?, ?, ?, ?, ?";
            OleDbCommand cmd = sqlDb.CreateCommand(strSql);

```

```

cmd.Parameters.Clear();
cmd.Parameters.AddWithValue("", firstname);
cmd.Parameters.AddWithValue("", lastname);
cmd.Parameters.AddWithValue("", address);
cmd.Parameters.AddWithValue("", tel);
cmd.Parameters.AddWithValue("", reposition);
cmd.Parameters.AddWithValue("", username);
cmd.Parameters.AddWithValue("", password);
cmd.Parameters.AddWithValue("", salary);
sqlDb.ExecuteReader(cmd);

sqlDb.CommitTrans();
}
catch(Exception ex)
{
sqlDb.RollbackTrans();
throw ex;
}
}

public void UpdateTechnicianProfile(int id, string firstname, string
lastname, string address, string tel, int reposition, string username,
string password, decimal salary)
{
using(var sqlDb = new SqlDb())
{
try
{
sqlDb.BeginTrans();
const string strSql = "exec UpdateTechnicalProfile ?,?,?,?, ?, ?, ?, ?, ?";
OleDbCommand cmd = sqlDb.CreateCommand(strSql);
cmd.Parameters.Clear();
cmd.Parameters.AddWithValue("", id);
cmd.Parameters.AddWithValue("", firstname);
cmd.Parameters.AddWithValue("", lastname);
cmd.Parameters.AddWithValue("", address);
cmd.Parameters.AddWithValue("", tel);
cmd.Parameters.AddWithValue("", reposition);
cmd.Parameters.AddWithValue("", username);
cmd.Parameters.AddWithValue("", password);
cmd.Parameters.AddWithValue("", salary);
sqlDb.ExecuteReader(cmd);
sqlDb.CommitTrans();
}
catch(Exception ex)
{
sqlDb.RollbackTrans();
throw ex;
}
}
}

```

```

public void InsertEquipment(string code, string name, string building,
string floor, string room, string detailsvendor, DateTime? installdate,
string imagepath, int? lifetime, int Ref_DefectedStuff)
{
    using (var sqlDb = new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            const string strSql = "exec InsertEquipmentDetail ?,?,?,?,?,?";
            OleDbCommand cmd = sqlDb.CreateCommand(strSql);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", code);
            cmd.Parameters.AddWithValue("", name);
            cmd.Parameters.AddWithValue("", building);
            cmd.Parameters.AddWithValue("", floor);
            cmd.Parameters.AddWithValue("", room);
            cmd.Parameters.AddWithValue("", detailsvendor);
            cmd.Parameters.AddWithValue("", installdate);
            cmd.Parameters.AddWithValue("", imagepath);
            cmd.Parameters.AddWithValue("", lifetime);
            cmd.Parameters.AddWithValue("", Ref_DefectedStuff);
            sqlDb.ExecuteReader(cmd);
            sqlDb.CommitTrans();
        }
        catch (Exception ex)
        {
            sqlDb.RollbackTrans();
            throw ex;
        }
    }
}

public void UpdateEquipment(string code, string name, string building,
string floor, string room, string detailsvendor, DateTime installdate, string
imagepath, int lifetime, int Ref_DefectedStuff)
{
    using (var sqlDb = new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            const string strSql = "exec UpdateEquipmentDetail ?,?,?,?,?";
            OleDbCommand cmd = sqlDb.CreateCommand(strSql);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", code);
            cmd.Parameters.AddWithValue("", name);
            cmd.Parameters.AddWithValue("", building);
            cmd.Parameters.AddWithValue("", floor);
            cmd.Parameters.AddWithValue("", room);
            cmd.Parameters.AddWithValue("", detailsvendor);
            cmd.Parameters.AddWithValue("", installdate);
            cmd.Parameters.AddWithValue("", imagepath);
            cmd.Parameters.AddWithValue("", lifetime);

```

```

        cmd.Parameters.AddWithValue("", Ref_DefectedStuff);
        sqlDb.ExecuteReader(cmd);
        sqlDb.CommitTrans();
    }
    catch(Exception ex)
    {
        sqlDb.RollbackTrans();
        throw ex;
    }
}

public void InsertMaintenanceJob(string maintenanceCode, string firstname,
string lastname, string refcode, decimal hours, string details, string
equipmentcode, int defected, int? technicianid, string tel, string createby,
string modifyby, string jobtype, int Ref_MaintenanceStatusId, DateTime
createdate)
{
    using(var sqlDb = new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            const string strSql = "exec InsertMaintenanceJob
?,?,?,?,?,?,?,?,?,?,?,?,?";
            OleDbCommand cmd = sqlDb.CreateCommand(strSql);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", maintenanceCode);
            cmd.Parameters.AddWithValue("", firstname);
            cmd.Parameters.AddWithValue("", lastname);
            cmd.Parameters.AddWithValue("", refcode);
            cmd.Parameters.AddWithValue("", hours);
            cmd.Parameters.AddWithValue("", details);
            cmd.Parameters.AddWithValue("", equipmentcode);
            cmd.Parameters.AddWithValue("", defected);
            cmd.Parameters.AddWithValue("", technicianid);
            cmd.Parameters.AddWithValue("", tel);
            cmd.Parameters.AddWithValue("", createby);
            cmd.Parameters.AddWithValue("", modifyby);
            cmd.Parameters.AddWithValue("", jobtype);
            cmd.Parameters.AddWithValue("", Ref_MaintenanceStatusId);
            cmd.Parameters.AddWithValue("", createdate.Year.ToString()+"-"+
createdate.Month.ToString()+"-"+createdate.Day.ToString()+" "+createdate.Hour
+" "+createdate.Minute +" "+createdate.Second);
            sqlDb.ExecuteReader(cmd);

            sqlDb.CommitTrans();
        }
        catch(Exception ex)
        {
            sqlDb.RollbackTrans();
            throw ex;
        }
    }
}

```

```

    }
}
public void UpdateMaintenanceJob(string maintenanceCode, string firstname,
string lastname, string refcode, decimal hours, string details, string
equipmentcode, int defected, int? positionid, string tel, int?
maintenanceStatus, string followup, string jobtype)
{
    using(var sqlDb = new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            const string strSql = "exec UpdateMaintenanceJob
            ?,?,?,?, ?, ?, ?, ?, ?, ?, ?, ?";
            OleDbCommand cmd = sqlDb.CreateCommand(strSql);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", maintenanceCode);
            cmd.Parameters.AddWithValue("", firstname);
            cmd.Parameters.AddWithValue("", lastname);
            cmd.Parameters.AddWithValue("", refcode);
            cmd.Parameters.AddWithValue("", hours);
            cmd.Parameters.AddWithValue("", details);
            cmd.Parameters.AddWithValue("", equipmentcode);
            cmd.Parameters.AddWithValue("", defected);
            cmd.Parameters.AddWithValue("", positionid);
            cmd.Parameters.AddWithValue("", tel);
            cmd.Parameters.AddWithValue("", maintenanceStatus);
            cmd.Parameters.AddWithValue("", followup);
            cmd.Parameters.AddWithValue("", jobtype);
            sqlDb.ExecuteReader(cmd);
            sqlDb.CommitTrans();
        }
        catch(Exception ex)
        {
            sqlDb.RollbackTrans();
        }
        throw ex;
    }
}

public void InsertMaintenancePlanDetails(SqlDb sqlDb, string Details, int
day, DateTime createdate, DateTime nextdate, int? reftechnic, string
refequipment, int? HoursForFixed)
{
    const string strSql = "exec [InsertMaintenancePlanDetail] ?, ?, ?, ?, ?, ?";
    OleDbCommand cmd = sqlDb.CreateCommand(strSql);
    cmd.Parameters.Clear();
    cmd.Parameters.AddWithValue("", Details);
    cmd.Parameters.AddWithValue("", day);
    cmd.Parameters.AddWithValue("", createdate);
    cmd.Parameters.AddWithValue("", nextdate);
    cmd.Parameters.AddWithValue("", reftechnic);
    cmd.Parameters.AddWithValue("", refequipment);
}

```



```

        cmd.Parameters.AddWithValue("", HoursForFixed);
        sqlDb.ExecuteReader(cmd);
    }
    public void UpdateMaintenancePlanAllDetail(SqlDb sqlDb, int reposition,
string refequipmentcode, string detail, int id, int duration, int?
HoursForFixed)
    {
        const string strSql = "exec [UpdateMaintenancePlanAllDetail]?, ?, ?, ?, ?, ?";
        OleDbCommand cmd = sqlDb.CreateCommand(strSql);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", id);
        cmd.Parameters.AddWithValue("", detail);
        cmd.Parameters.AddWithValue("", duration);
        cmd.Parameters.AddWithValue("", reposition);
        cmd.Parameters.AddWithValue("", refequipmentcode);
        cmd.Parameters.AddWithValue("", HoursForFixed);
        sqlDb.ExecuteReader(cmd);
    }
    public void DeleteMaintenancePlanDetail(SqlDb sqlDb, string equipmentcode)
    {
        const string strSql = "exec [DeleteMaintenancePlanDetail]?";
        OleDbCommand cmd = sqlDb.CreateCommand(strSql);
        cmd.Parameters.Clear();
        cmd.Parameters.AddWithValue("", equipmentcode);
        sqlDb.ExecuteReader(cmd);
    }
    public void InsertMaintenancePlanDetails(int reftechnician, string
refequipmentcode, DateTime maintenanceDate,
List<_PlanMaintenanceDetailStruct> details)
    {
        using (var sqlDb = new SqlDb())
        {
            try
            {
                sqlDb.BeginTrans();
                foreach (var i in details)
                {
                    var obj = new PlanMaintenanceDetailStruct()
                    {
                        Details = i.Details,
                        Freq = i.Freq,
                        MaintenanceDate = maintenanceDate,
                        NextMaintenanceDate = maintenanceDate.AddDays(i.Freq)
                        ,HoursForFixed = i.HoursForFixed
                    };
                    InsertMaintenancePlanDetails(sqlDb, obj.Details,
obj.Freq, obj.MaintenanceDate,
obj.NextMaintenanceDate.Value, reftechnician, refequipmentcode, obj.HoursForFixed);
                }
                sqlDb.CommitTrans();
            }
            catch (Exception ex)

```

```

        {
            sqlDb.RollbackTrans();
        }
        throw ex;
    }
}

public void UpdateMaintenancePlanDetails(int reposition, string
refequipmentcode, DateTime maintenanceDate,
List<_PlanMaintenanceDetailStruct> details)
{
    using(var sqlDb = new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            DeleteMaintenancePlanDetail(sqlDb, refequipmentcode);

            foreach(var i in details)
            {
                var obj = new PlanMaintenanceDetailStruct()
                {
                    Details = i.Details,
                    Freq = i.Freq,
                    MaintenanceDate = maintenanceDate.AddDays(i.Freq),
                    NextMaintenanceDate = maintenanceDate.AddDays(i.Freq
* 2)
                    , Ref_EquipmentCode = refequipmentcode
                    , Ref_TechnicalId = reposition
                    , HoursForFixed = i.HoursForFixed
                };
                InsertMaintenancePlanDetails(sqlDb, obj.Details,
obj.Freq, obj.MaintenanceDate, obj.NextMaintenanceDate.Value, reposition,
refequipmentcode, obj.HoursForFixed);
            }
            sqlDb.CommitTrans();
        }
        catch(Exception ex)
        {
            sqlDb.RollbackTrans();

            throw ex;
        }
    }
}

public void UpdateMaintenancePlanDetail(DateTime date)
{
    using(var sqlDb = new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            const string strSql = "exec UpdateMaintenancePlanDetail ?";
            OleDbCommand cmd = sqlDb.CreateCommand(strSql);
            cmd.Parameters.Clear();

```

```

        cmd.Parameters.AddWithValue("", date.Year.ToString()+"-"+
date.Month.ToString()+"-"+date.Day.ToString());
        sqlDb.ExecuteReader(cmd);
        sqlDb.CommitTrans();
    }
    catch(Exception ex)
    {
        sqlDb.RollbackTrans();
        throw ex;
    }
}
}
public void DeletePlanMaintenanceByEquipmentcode(string code)
{
    using(var sqlDb =new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            const string strSql ="exec DeletePlanMaintenanceByEquipmentCode ?";
            OleDbCommand cmd =sqlDb.CreateCommand(strSql);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", code);
            sqlDb.ExecuteReader(cmd);
            sqlDb.CommitTrans();
        }
        catch(Exception ex)
        {
            sqlDb.RollbackTrans();
            throw ex;
        }
    }
}
public void DeletePlanMaintenanceByDetailId(int id)
{
    using(var sqlDb =new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            const string strSql ="exec DeletePlanMaintenanceByDetailId ?";
            OleDbCommand cmd =sqlDb.CreateCommand(strSql);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", id);
            sqlDb.ExecuteReader(cmd);
            sqlDb.CommitTrans();
        }
        catch(Exception ex)
        {
            sqlDb.RollbackTrans();
            throw ex;
        }
    }
}
}

```

```

public void InsertUserInfo(string name, string lastname, string building,
string floor, string room, string tel, string email, string username, string
password)
{
    using (var sqlDb = new SqlDb())
    {
        try
        {
            sqlDb.BeginTrans();
            const string strSql = "exec InsertUserInfo ?,?,?,?, ?, ?, ?, ?, ?";
            OleDbCommand cmd = sqlDb.CreateCommand(strSql);
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("", name);
            cmd.Parameters.AddWithValue("", lastname);
            cmd.Parameters.AddWithValue("", building);
            cmd.Parameters.AddWithValue("", floor);
            cmd.Parameters.AddWithValue("", room);
            cmd.Parameters.AddWithValue("", tel);
            cmd.Parameters.AddWithValue("", email);
            cmd.Parameters.AddWithValue("", username);
            cmd.Parameters.AddWithValue("", password);
            sqlDb.ExecuteReader(cmd);
            sqlDb.CommitTrans();
        }
        catch (Exception ex)
        {
            sqlDb.RollbackTrans();
            throw ex;
        }
    }
}
//Report
public List<ReportPieStruct> ReportDailyJob()
{
    string strSql = "exec [ReportDailyJob]";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSql);
        DataTable dt = sqlDB.GetDataTable(cmd);
        var list = new List<ReportPieStruct>();
        list.AddRange(from DataRow dr in dt.Rows select new
ReportPieStruct(dr));
        return list;
    }
}

public List<ReportPieStruct> ReportDailyPlan()
{
    string strSql = "exec [ReportDailyPlan]";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSql);

```

```

        DataTable dt = sqlDB.GetDataTable(cmd);
        var list = new List<ReportPieStruct>();
        list.AddRange(from DataRow dr in dt.Rows select new
ReportPieStruct(dr));
        return list;
    }
}
public List<ReportPieStruct> ReportMonthlyJob()
{
    string strSQL = "exec [ReportMonthlyJob]";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        DataTable dt = sqlDB.GetDataTable(cmd);
        var list = new List<ReportPieStruct>();
        list.AddRange(from DataRow dr in dt.Rows select new
ReportPieStruct(dr));
        return list;
    }
}
public List<ReportPieStruct> ReportMonthlyPlan()
{
    string strSQL = "exec [ReportMonthlyPlan]";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        DataTable dt = sqlDB.GetDataTable(cmd);
        var list = new List<ReportPieStruct>();
        list.AddRange(from DataRow dr in dt.Rows select new
ReportPieStruct(dr));
        return list;
    }
}
public List<ReportPieStruct> ReportMonthlyJobforType()
{
    string strSQL = "exec [ReportMonthlyJobforType]";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        DataTable dt = sqlDB.GetDataTable(cmd);
        var list = new List<ReportPieStruct>();
        list.AddRange(from DataRow dr in dt.Rows select new
ReportPieStruct(dr));
        return list;
    }
}
public List<ReportPieStruct> ReportCountJobWithEquipment()
{
    string strSQL = "exec [ReportCountJobWithEquipment]";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        DataTable dt = sqlDB.GetDataTable(cmd);

```

```

    var list = new List<ReportPieStruct>();
        list.AddRange(from DataRow dr in dt.Rows select new
ReportPieStruct(dr));
    return list;
    }
}
public List<ReportPieStruct> ReportSummaryHoursForFixed()
{
    string strSQL = "exec [ReportSummaryHoursForFixed]";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        DataTable dt = sqlDB.GetDataTable(cmd);
        var list = new List<ReportPieStruct>();
            list.AddRange(from DataRow dr in dt.Rows select new
ReportPieStruct(dr));
        return list;
    }
}
public List<ReportPieStruct> ReportSummaryReward()
{
    string strSQL = "exec [ReportSummaryReward]";
    using (var sqlDB = new SqlDb())
    {
        OleDbCommand cmd = sqlDB.CreateCommand(strSQL);
        DataTable dt = sqlDB.GetDataTable(cmd);
        var list = new List<ReportPieStruct>();
            list.AddRange(from DataRow dr in dt.Rows select new
ReportPieStruct(dr));
        return list;
    }
}
}
}

```

7. ชุดคำสั่ง `SqlDb.cs`

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Data.OleDb;
using System.Data;
using System.Configuration;
namespace Data.OleDb
{
    public class SqlDb: IDisposable
    {
        //Fields
        private OleDbDataAdapter adp;
        private OleDbConnection cnn;
        private IsolationLevel IsoLevel;
        private string strConnectionString;
        private OleDbTransaction trn;

        private string _ConnectionString = string.Empty;
        //Methods
        public SqlDb()
        {
            this.IsoLevel = IsolationLevel.Unspecified;
            this.strConnectionString = "";
            try
            {
                _ConnectionString =
ConfigurationManager.ConnectionStrings["MaintenanceDB"].ConnectionString;
                string connectionString = _ConnectionString;
                this.cnn = new OleDbConnection(connectionString);
            }
            catch(Exception exception)
            {
                throw new Exception("SqlDb.SqlDb():" + exception.Message);
            }
        }
        public SqlDb(string pstrConnectionString)
        {
            this.IsoLevel = IsolationLevel.Unspecified;
            this.strConnectionString = "";
            this.strConnectionString = pstrConnectionString;
            try
            {
                this.cnn = new OleDbConnection(this.strConnectionString);
            }
            catch(Exception exception)
            {
                throw new Exception("SqlDb.SqlDb(string):" + exception.Message);
            }
        }
        public void BeginTrans()
        {
            try

```

```

        {
            if(this.cnn.State == ConnectionState.Closed)
            {
                this.cnn.Open();
            }
            this.trn = this.cnn.BeginTransaction(this.IsoLevel);
        }
        catch(Exception exception)
        {
            throw new Exception("SqlDb.BeginTrans0:" + exception.Message);
        }
    }
    public void CommitTrans()
    {
        try
        {
            try
            {
                this.trn.Commit();
                if(this.cnn.State == ConnectionState.Open)
                {
                    this.cnn.Close();
                }
            }
            catch(Exception exception)
            {
                throw new Exception("SqlDb.CommitTrans0:" + exception.Message);
            }
        }
        finally
        {
            this.trn = null;
            this.cnn.Close();
        }
    }
    public OleDbCommand CreateCommand()
    {
        if(this.Transaction == null)
        {
            return new OleDbCommand("", new OleDbConnection(_ConnectionString));
        }
        return new OleDbCommand("", this.cnn, this.Transaction);
    }
    public OleDbCommand CreateCommand(string StrSQL)
    {
        if(this.Transaction == null)
        {
            return new OleDbCommand(StrSQL, new OleDbConnection(_ConnectionString));
        }
        return new OleDbCommand(StrSQL, this.cnn, this.Transaction);
    }
    public void Dispose()
    {
        this.adp = null;
        this.cnn = null;
    }

```



```

this.strConnectionString = null;
this.trn = null;
}
public void ExecuteCommand(OleDbCommand cmd)
{
try
{
cmd.ExecuteNonQuery();
}
catch(Exception exception)
{
throw new Exception("SqlDb.Execute(string):"+exception.Message);
}
}
public string ExecuteReader(OleDbCommand cmd)
{
OleDbDataReader reader = null;
string str;
try
{
try
{
reader = cmd.ExecuteReader();
if (reader.Read())
{
string str2 = reader.GetValue(0).ToString();
reader.Close();
return str2;
}
str = "";
}
catch(Exception exception)
{
throw new Exception("SqlDb.Execute(string):"+exception.Message);
}
}
finally
{
reader.Close();
}
return str;
}
public object ExecuteScalarCommand(OleDbCommand cmd)
{
object obj2;
try
{
obj2 = cmd.ExecuteScalar();
}
catch(Exception exception)
{
throw new Exception("SqlDb.Execute(string):"+exception.Message);
}
return obj2;
}
}

```

```

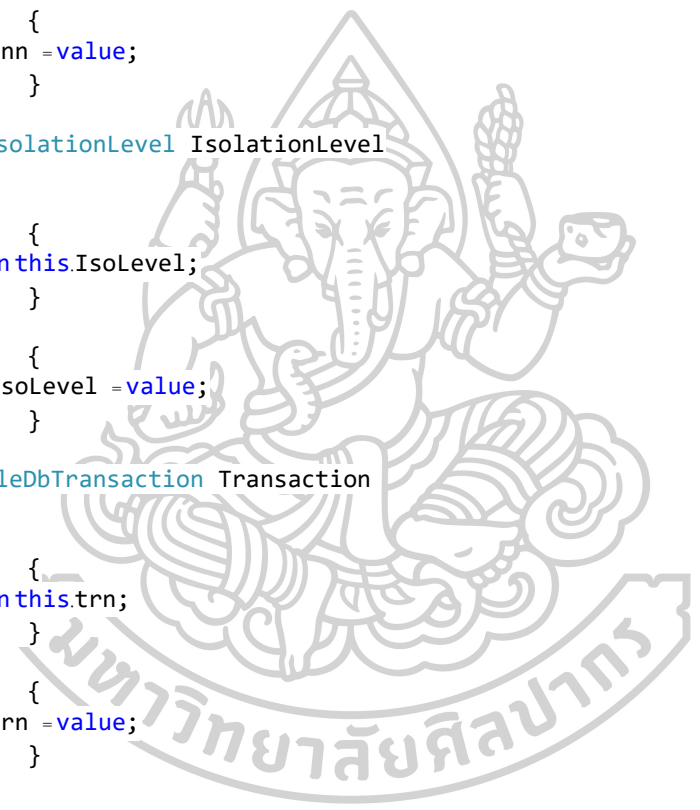
public DataRow GetDataRow(OleDbCommand cmd)
{
    DataRow row;
    try
    {
        this.adp = null;
        this.adp = new OleDbDataAdapter(cmd);
        DataSet dataSet = new DataSet();
        this.adp.Fill(dataSet);
        if (dataSet.Tables[0].Rows.Count == 0)
        {
            return null;
        }
        row = dataSet.Tables[0].Rows[0];
    }
    catch (Exception exception)
    {
        throw new Exception("SqlDb.Query2DataRow(string):" + exception.Message);
    }
    return row;
}

public DataTable GetDataTable(OleDbCommand cmd)
{
    DataTable table;
    try
    {
        this.adp = null;
        this.adp = new OleDbDataAdapter(cmd);
        DataSet dataSet = new DataSet();
        this.adp.Fill(dataSet);
        table = dataSet.Tables[0];
    }
    catch (Exception exception)
    {
        throw new Exception("SqlDb.Query2DataTable(string):" + exception.Message + "");
    }
    return table;
}

public void RollbackTrans()
{
    try
    {
        try
        {
            this.trn.Rollback();
            if (this.cnn.State == ConnectionState.Open)
            {
                this.cnn.Close();
            }
        }
        catch (Exception exception)
        {
            throw new Exception("SqlDb.RollbackTrans():" + exception.Message);
        }
    }
}

```

```
finally
    {
        this.trn = null;
        this.cnn.Close();
    }
}
//Properties
public OleDbConnection Connection
    {
        get
            {
                return this.cnn;
            }
        set
            {
                this.cnn = value;
            }
    }
public IsolationLevel IsolationLevel
    {
        get
            {
                return this.IsolationLevel;
            }
        set
            {
                this.IsolationLevel = value;
            }
    }
public OleDbTransaction Transaction
    {
        get
            {
                return this.trn;
            }
        set
            {
                this.trn = value;
            }
    }
}
}
```



## รายการอ้างอิง

1. FRANK, B., *FACILITIES MANAGEMENT HANDBOOK*, ed. FOURTH. 2009, USA: Elsevier Ltd.
2. Barrie , C. and S. Peter *BUILDING MAINTENANCE MANAGEMENT*, ed. Second. 2007, USA Blackwell.
3. วิจิตรบุษบา มารมย์, การบริหารจัดการทรัพยากรอาคาร *Journal of Architectural Research and Studies*, 2002. **1**: p. 225 - 236.
4. ชีรพล ต้นสัจจา, กรณีศึกษาเรื่องระบบการจัดการบำรุงรักษาเครื่องปรับอากาศชนิด *Package Water Cooled* ในการบริหารอาคาร, in ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์. 2544, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี.
5. นางสุมาลี ชีระวันทนี, การศึกษาข้อมูลสำหรับการจัดการอาคารและทัศนคติการบริหารจัดการอาคารสูงของผู้บริหารอาคาร, in สาขาวิชาวิศวกรรมและการบริหารการก่อสร้าง. 2547, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี.
6. ทรงภพ บุรณะศิลป์, การพัฒนาโปรแกรมการจัดการระบบงานบำรุงรักษาด้วยคอมพิวเตอร์ กรณีศึกษา ระบบบำบัดน้ำเสีย ในโรงงานผลิตมะหมี่กึ่งสำเร็จรูป, in สาขาวิชาวิศวกรรมการจัดการอุตสาหกรรม ภาควิชาวิศวกรรมอุตสาหกรรม. 2550, สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.
7. ธนากร ณ พัทลุง, *CMMS*เพื่อระบบการจัดการงานบำรุงรักษาที่ดี. ไฟฟ้าอุตสาหกรรม, 2553. **4**: p. 70 – 72.
8. *CMMS Maintenance Management Software*. 20 มีนาคม 2561]; Available from: <https://fastmaint-cmms-maintenance-management-software.soft32.com/screenshots/>.
9. มณีโชติ สมานไทย, การเขียนโค้ด *ASP.NET* : ฉบับสมบูรณ์. 2546: อินโฟเพรส.
10. *CAFM Explorer* [cited 2018 เมษายน 2561]; Available from: <http://www.cafmexplorer.com>.
11. *digitalmarketplace*. เมษายน 2561]; Available from: <https://www.digitalmarketplace.service.gov.uk/g-cloud/services/276994709056447>.
12. *Work Order Management*. [cited 2018 เมษายน 2561]; Available from: <https://www.hippocmms.com>.
13. ASPx controls. *ASPxGridView*. 2018 [cited 2018 April 2]; Available from: <https://www.devexpress.com/Support/Center/>.

14. Google chart. *Google chart tools*. 2018 [cited 2018 April 25]; Available from: <https://google-developers.appspot.com/chart/>
15. Autocomplete. *jqueryui*. 2018 [cited 2018 April 2]; Available from: <https://jqueryui.com/autocomplete/>.





## ประวัติผู้เขียน

ชื่อ-สกุล	มาย ยินชัย
วัน เดือน ปี เกิด	6 กรกฎาคม 2523
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	พศ. 2546 วศบ. วิศวกรรมคอมพิวเตอร์ วิศวกรรมศาสตรบัณฑิต มหาวิทยาลัยสยาม
ที่อยู่ปัจจุบัน	10 ถ.โพธิ์ไกรร้ง ต.ท่าราบ อ.เมือง จ.เพชรบุรี 76000

