



โปรแกรมแสดงผลและควบคุมสถานะการผลิต
PROGRAM DATA DISPLAY MANUFACTURING CONTROL

พรประสิทธิ์ ศุภพล 6052500221

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
คณะวิศวกรรมศาสตร์และเทคโนโลยี สถาบันการจัดการปัญญาภิวัฒน์
ปีการศึกษา พ.ศ. 2563
ลิขสิทธิ์เป็นของสถาบันการจัดการปัญญาภิวัฒน์




โปรแกรมแสดงผลและควบคุมสถานะการผลิต
PROGRAM DATA DISPLAY MANUFACTURING CONTROL

พรประสิทธิ์ ศุภพล 6052500221

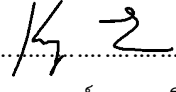
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
คณะวิศวกรรมศาสตร์และเทคโนโลยี สถาบันการจัดการปัญญาภิวัฒน์
ปีการศึกษา พ.ศ. 2563
ลิขสิทธิ์เป็นของสถาบันการจัดการปัญญาภิวัฒน์

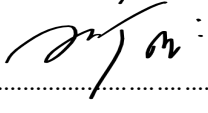
เรื่อง	โปรแกรมแสดงผลและควบคุมสถานการณ์ผลิต Program Data Display Manufacturing Control.
ชื่อผู้วิจัย	นายพรประสิทธิ์ ศุภพล
ชื่ออาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.เกรียงไกร ทศนวิภาส
ชื่อปริญญา	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
ปีการศึกษา	2564

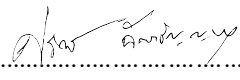
ได้รับการอนุมัติเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ



..... คณบดีคณะวิศวกรรมศาสตร์และเทคโนโลยี
(รองศาสตราจารย์ ดร. พิชิต ชาญเกียรติก้อง)

คณะกรรมการสอบป้องกันวิทยานิพนธ์


..... ประธานกรรมการสอบ
(ผู้ช่วยศาสตราจารย์ ดร. เกรียงไกร ทศนวิภาส)


..... กรรมการ
(อาจารย์ วรวุฒิ กังหัน)


..... กรรมการ
(ดร. ศรัณย์ ฉัตรธัญญกิจ)


..... กรรมการ
(ดร. ภาคภูมิ ปฐมภาคย์)

ชื่อเรื่อง	โปรแกรมแสดงผลและควบคุมสถานการณ์ผลิต
ชื่อผู้วิจัย	นายพรประสิทธิ์ ศุภพล
ชื่ออาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.เกรียงไกร ทศนวิภาส
ชื่อปริญญา	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
ปีการศึกษา	2564
หน้า	

บทคัดย่อ

เนื่องจากในปัจจุบันสายการผลิตรถยนต์ในบริษัท อาร์เอ็มเอ ออโตโมทีฟ จำกัด นั้นยังคงมีการแสดงข้อมูลการผลิตรถยนต์ต่อวันโดยการใช้ป้ายแขวนและกระดาน ซึ่งบอกข้อมูลไม่ครบถ้วนและยากต่อการมองเห็น ดังนั้น ผู้วิจัยจึงได้สร้างโปรแกรมสำหรับแสดงและจัดเก็บข้อมูลการผลิตต่อวัน โดยจะมีการแบ่งส่วนในการเก็บข้อมูลและแสดงผลในรูปแบบแดชบอร์ด (Dashboard) และส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface:GUI) เพื่อให้ง่ายต่อการตรวจสอบสถานะและสามารถนำข้อมูลจากฐานข้อมูล (Database) มาวิเคราะห์ย้อนหลังโดยมีรูปแบบการจัดเก็บเป็นไฟล์นามสกุล .xls

โดยเมื่อมีการติดตั้งหน้าจอแสดงผลการผลิตดังกล่าวแล้ว ที่วิศวกรและพนักงานในสายการผลิต สามารถตรวจสอบเวลามาตรฐานและสถานะการทำงานในไลน์การผลิตได้ง่าย รวมถึงสามารถรู้ข้อมูลการเกิด การหยุดการทำงานของสายการผลิตในสถานี่ทำงานได้อย่างรวดเร็ว และง่ายต่อการนำข้อมูลต่าง ๆ มาวิเคราะห์ปัญหาที่เกิดขึ้น พบว่าสามารถลดระยะเวลาในการค้นหาข้อมูลที่จัดเก็บบนกระดานของพนักงานจากเดิมประมาณ 30 นาที เหลือ 2 นาที คิดเป็นอัตราส่วน 30:2 เท่า เมื่อเทียบกับการจัดเก็บโดยกระดาน

Title	Program data display manufacturing control
Author's Name	Mr. Phomprasit Suphaphon
Advisor	Asst. Prof. Dr. Kiangkrai Tassanavipas
Degree	Bachelor of Engineering (Robotics and Automation Engineering)
Academic Year	2020
Page	

Abstract

Nowadays automotive productive lines of RMA Automotive company (limited) using paper and whiteboard for showing the data but that does not have complete and difficult to see. Therefore, this thesis creating a program for display productive data per day using the graphical user interface (GUI) form. So that can eliminate the problem and comfortable for checking and analyzing data Which Stores files in .xls extension format.

When installed the Display screen on productive lines, the program result showed a reduction in search time compared to paper searches from 30 minutes to 2 minutes or a reduction in the ratio of 30:2.

กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.เกรียงไกร ทศนวิภาส ที่ปรึกษาปริญญานิพนธ์ ที่กรุณาให้คำปรึกษา และคำแนะนำ แนวคิดในการทำเล่มนี้ ตลอดจนตรวจสอบและแก้ไขปริญญานิพนธ์ อย่างสมบูรณ์

ขอขอบพระคุณ ดร.ภาคภูมิ ปฐมภาคย์ ดร.ศรัณย์ ฉัตรธัญญกิจ และอาจารย์วรวิทย์ กังหัน ที่ได้กรุณาเป็นกรรมการสอบปริญญานิพนธ์ ตลอดจนตรวจสอบและแก้ไขปริญญานิพนธ์ อย่างสมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณ คณาจารย์ บุคลากร และนักศึกษา สาขาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติทุกคนที่ให้คำปรึกษาและกำลังในการทำเล่มนี้

และสุดท้าย ข้าพเจ้าขอโน้มรำลึกถึงพระคุณของบิดา มารดา และครอบครัว ที่ส่งเสริมและสนับสนุนในทุกๆเรื่อง จนสำเร็จการศึกษา

พรประสิทธิ์ ศุภพล

พฤษภาคม 2564

สารบัญ

หน้า

บทคัดย่อ.....	ก
Abstract.....	ข
กิตติกรรมประกาศ.....	ข
สารบัญ.....	ง
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	1
1.3 สมมติฐานของงานวิจัย	1
1.4 ขอบเขตของการวิจัย	2
1.5 ข้อจำกัดของการวิจัย	2
1.6 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย	2
1.7 แผนการดำเนินการ	2
บทที่ 2 วรรณกรรมที่เกี่ยวข้อง.....	3
2.1 ระบบฐานข้อมูล.....	3
2.1.2 องค์ประกอบของระบบฐานข้อมูล.....	5
2.1.3 เครื่องมือที่ใช้ในการวิเคราะห์และออกแบบระบบฐานข้อมูล.....	6
2.1.3 การออกแบบฐานข้อมูลด้วย ER-Diagram.....	10
2.2 โปรแกรมที่ทำหน้าที่ให้บริการ Database.....	13
2.2.1 มายเอสคิวแอล (MySQL).....	13
2.2.2 พีเอชพีมายแอดมิน (phpMyAdmin).....	15
2.3 Microsoft Visual Studio.....	16

2.3.1	ดอตเน็ตเฟรมเวิร์ค (.NET Framework)	16
2.4.	ภาษาซีชาร์ป (C# Programming Language)	18
2.4.1	โครงสร้างโปรแกรมภาษาซีชาร์ป	18
2.5	แนวคิดเชิงวัตถุ (Object-Oriented Concept)	20
2.5.1	แนวคิดเชิงวัตถุ (Object-Oriented)	20
2.5.2	ออบเจกต์ (Object)	20
2.5.3	เอนแคปซูเลชัน (Encapsulation)	20
2.5.4	คลาส (Class)	20
2.5.5	การสืบทอดคุณสมบัติ (Inheritance)	21
2.5.6	ความสัมพันธ์ระหว่างคลาส (Class Relationship)	21
2.5.7	การพัฒนาระบบเชิงวัตถุ	22
2.5.8	แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented Database Model)	23
2.5.9	ยูเอ็มแอล (UML)	23
2.6	การควบคุมด้วยการมองเห็น (Visual Control)	24
2.7	ประสบการณ์ของผู้ใช้งาน และ ส่วนต่อประสานกับผู้ใช้	26
บทที่ 3	วิธีดำเนินการวิจัย	28
3.1	ประชากรที่ใช้ในการศึกษา	28
3.1.2	เครื่องมือที่ใช้ในงานวิจัย	28
3.2	การศึกษาและรวบรวมข้อมูล	29
3.2.1	การศึกษาและรวบรวมข้อมูลสายการประกอบอาชีพที่ดี	29
3.3	เสนอแนวทางการติดตั้ง	30
3.4	วิเคราะห์และออกแบบระบบ	31
3.4.1.	การออกแบบแผนภาพแสดงการทำงานของโปรแกรมแสดงผลและควบคุมสถานการณ์ผลิต	31

3.4.2 การออกแบบระบบฐานข้อมูล.....	34
3.4.3 การพัฒนาระบบฐานข้อมูลโปรแกรมแสดงผลและควบคุมสถานการณ์ผลิต.....	34
3.4.4 ความสัมพันธ์ของตาราง.....	38
3.5 การออกแบบหน้าจอแสดงผลข้อมูล.....	42
3.5.1 การแสดงผลข้อมูลในสายการผลิต.....	42
3.5.2 การควบคุม.....	44
3.5.3 การแสดงข้อมูลแจ้งเตือนผ่านแอปพลิเคชันไลน์.....	45
บทที่ 4 ผลการวิจัย.....	47
4.1 ผลการดำเนินงานวิจัย	47
4.1.1 ผลของการสำรวจของประชากร 5 คน ที่เข้าใช้โปรแกรมแสดงผลและควบคุมสถานการณ์การผลิต.....	47
4.1.2 การเปรียบเทียบการทำงานแบบเดิมและแบบใหม่.....	48
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ.....	50
5.1 สรุปผลการดำเนินงาน.....	50
5.2 ประโยชน์ของวิจัย.....	50
5.3 ปัญหาที่พบ.....	50
5.5 ข้อเสนอแนะ.....	51
บรรณานุกรม	52
ภาคผนวก.....	54
โค้ดทั้งหมดของผู้วิจัยนี้.....	55
1.โค้ดหน้าเข้าสู่ระบบ.....	55
2. โค้ดหน้าการทำงานหลัก	58
3. โค้ดหน้าป้อนข้อมูลการผลิตรถยนต์.....	66
4. โค้ดหน้าป้อนข้อมูลการเกิดเบรกดาว.....	95

5. ใค้ดหน้าป้อนข้อมูลการพักเบรก.....	111
6. ใค้ดหน้าแสดงผลในสายการผลิต.....	113
7. ใค้ดหน้าแสดงผลสถานะสายการผลิต.....	116
8. ใค้ดหน้าแสดงประวัติการเกิดเบรกดาว.....	123
9. ใค้ดหน้าแสดงเวลาพักเบรก.....	126

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 1.1 แผนการดำเนินการ	2
ตารางที่ 1.2 แผนการดำเนินงานช่วงที่ 2	2
ตารางที่ 2.1 คำสั่งที่ใช้สำหรับการฐานข้อมูล	14
ตารางที่ 3.1 แสดงแผนภาพการไหลของข้อมูล	34
ตารางที่ 3.2 ข้อมูลการผลิตรถยนต์ต่อวัน	40
ตารางที่ 3.3 ข้อมูลการหยุดทำงานของสายการผลิต	40
ตารางที่ 3.4 ข้อมูลการหยุดทำงานของสายการผลิต (ต่อ).....	41
ตารางที่ 3.5 ข้อมูลเวลาการพัก	41
ตารางที่ 3.6 ข้อมูลผู้ใช้งาน.....	41
ตารางที่ 4.1 แสดงคะแนนประชากร 5 คน ที่ใช้งานโปรแกรมแสดงผลและควบคุมสถานะการผลิต.47	
ตารางที่ 4.2 แสดงการเปรียบเทียบการทำงานแบบเดิมและแบบใหม่.....	48
ตารางที่ 4.3 แสดงระยะเวลาเปรียบเทียบการทำงานแบบเดิมและแบบใหม่	49

สารบัญภาพ

ภาพ	หน้า
ภาพที่ 2.1 โครงสร้างข้อมูล Data Structure.....	4
ภาพที่ 2.2 สัญลักษณ์ที่ใช้ในการเขียนแผนภาพกระแสข้อมูล.....	6
ภาพที่ 2.3 แสดง Context diagram: DFD Level 0.....	7
ภาพที่ 2.4 แสดง Data Flow Diagram Level 1.....	8
ภาพที่ 2.5 แสดง Data Flow Diagram Level 2.....	9
ภาพที่ 2.6 สัญลักษณ์ E-R Model	10
ภาพที่ 2.7 ภาพตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....	12
ภาพที่ 2.8 ภาพตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....	12
ภาพที่ 2.9 ภาพตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....	12
ภาพที่ 2.10 ตัวอย่างโปรแกรม phpMyAdmin.....	15
ภาพที่ 2.11 ดอตเน็ตเฟรมเวิร์ค (.NET Framework)	17
ภาพที่ 2.12 โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน.....	18
ภาพที่ 2.13 ตัวอย่างแสดงข้อความ Hello C# ออกทางจอภาพ.....	19
ภาพที่ 2.14 ตัวอย่างแสดงบอร์ดสื่อสารในสายการผลิต.....	25
ภาพที่ 2.15 ประสบการณ์ของผู้ใช้งาน และ ส่วนต่อประสานกับผู้ใช้.....	26
ภาพที่ 3.1 กระดานบอกสถานการณ์ผลิตต่อวัน	29
ภาพที่ 3.2 ป้ายแขวนบอกตัวเลข Lot และยอดการผลิตในเวลาปัจจุบัน	29
ภาพที่ 3.3 เช็ดยอดการผลิตตามหมายเลขประจำรถยนต์	30
ภาพที่ 3.4 ระบบควบคุมการแสดงผลภายในสายการผลิต.....	30
ภาพที่ 3.5 แผนภาพแสดงขั้นตอนการทำงานหน้า MFG Input.....	31
ภาพที่ 3.6 แผนภาพแสดงขั้นตอนการทำงานหน้า Breakdown Input	32
ภาพที่ 3.7 แผนภาพแสดงลำดับหน้าแสดงผลในสายการผลิต.....	33
ภาพที่ 3.8 แผนภาพบริบท Context Diagram.....	35

ภาพที่ 3.9 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1) ในส่วนของข้อมูล รถยนต์ในการผลิต	35
ภาพที่ 3.10 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1) ในส่วนของข้อมูล ข้อมูลการหยุดการทำงานของสายการผลิต	36
ภาพที่ 3.11 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1) ในส่วนของผู้ใช้ งานระบบ.....	36
ภาพที่ 3.12 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1) ในส่วนของเวลาพัก	36
ภาพที่ 3.13 แผนภาพแสดงความสัมพันธ์ของข้อมูล (Entity Relationship Diagram) ในส่วนของ ระบบจัดเก็บข้อมูลการแสดงผลและควบคุมสถานะการผลิต	37
ภาพที่ 3.14 แสดงความสัมพันธ์ระหว่าง ตารางประเภทผู้ใช้งานกับ ตารางระบบแสดงและควบคุมการ ผลิต.....	38
ภาพที่ 3.15 แสดงความสัมพันธ์ระหว่าง ตารางประเภทระบบแสดงและควบคุมการผลิต กับ ตาราง รถยนต์ที่ผลิตต่อวัน.....	38
ภาพที่ 3.16 ความสัมพันธ์ระหว่าง ตารางประเภทระบบแสดงและควบคุมการผลิต กับ ตารางการหยุด ทำงานของสายการผลิต.....	39
ภาพที่ 3.17 แสดงความสัมพันธ์ระหว่าง ตารางประเภทระบบแสดงและควบคุมการผลิต กับ ตารางเวลาพัก	39
ภาพที่ 3.18 ตัวอย่างหน้าแสดงผลการผลิต	42
ภาพที่ 3.19 ตัวอย่างหน้าแสดงสถานะสายการผลิต	42
ภาพที่ 3.20 ภาพการออกแบบหน้าแสดงผลประวัติการหยุดการทำงานของสายการผลิต.....	43
ภาพที่ 3.21 ตัวอย่างแสดงหน้าพักเบรก.....	43
ภาพที่ 3.22 ตัวอย่างหน้าป้อนข้อมูลการผลิต (MFG Input).....	44
ภาพที่ 3.23 ตัวอย่างหน้าป้อนข้อมูลการหยุดการทำงานของสายการผลิต (Breakdown Input).....	44
ภาพที่ 3.24 ตัวอย่างหน้ากำหนดเวลาการพักเบรก (Set Breaktime).....	45
ภาพที่ 3.25 ตัวอย่างการแจ้งเตือนการผลิตรถยนต์ต่อวันผ่านแอปพลิเคชันไลน์	45
ภาพที่ 3.26 ตัวอย่างหน้าป้อนข้อมูลการผลิต (MFG Input).....	46

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันสายการประกอบรถยนต์ ของบริษัท อาร์เอ็มเอ ออโตโมทีฟ จำกัด มีการดำเนินงานในรูปแบบสายการประกอบและดัดแปลงรถยนต์โดยเน้นการใช้แรงงานฝีมือของพนักงานเป็นหลัก ซึ่งในกระบวนการทำงานนั้นยอดการผลิตต่อวันยังคงใช้การแสดงผลโดยการแขวนป้ายบอกตัวเลขซึ่งยากต่อการมองเห็น และมีข้อมูลไม่ครบถ้วนยากต่อการติดตามผล

โดยตลอดระยะเวลาการฝึกงานตั้งแต่เดือนสิงหาคม 2563 ถึง เดือนกุมภาพันธ์ 2564 ผู้วิจัยได้เล็งเห็นถึงปัญหาด้านการควบคุมการผลิตด้วยการมองเห็น ดังนั้นจึงเป็นที่มาของปริญญานิพนธ์เรื่อง “โปรแกรมแสดงผลและควบคุมสถานะการผลิต” เพื่อเพิ่มประสิทธิภาพการควบคุมการผลิตโดยแสดงสถานะการผลิตต่อวันให้สามารถมองเห็นได้ง่าย ทั้งในส่วนของจำนวนนับ การเปรียบเทียบเป้าหมายต่อวัน หรือการแสดงผลรอบเวลามาตรฐานในการผลิต รวมถึงการเก็บฐานข้อมูล (Database) เมื่อเกิดการหยุดการทำงานของสายการผลิตเพื่อให้ง่ายต่อการนำข้อมูลมาวิเคราะห์เพื่อหาแนวทางและวิธีในการแก้ปัญหาต่อไป

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อเพิ่มประสิทธิภาพการควบคุมการผลิตโดยแสดงสถานะการผลิตต่อวัน

1.2.2 เพื่อแสดงเวลามาตรฐานและแสดงสถานะการทำงานในสายการผลิต ภายในแผนก Manufacturing Engineering

1.2.3 เพื่อเก็บรวบรวมข้อมูลการผลิตต่อวันรวมถึงการเกิดการหยุดการทำงานของสายการผลิต เพื่อให้ง่ายต่อการนำข้อมูลมาวิเคราะห์

1.3 สมมติฐานของงานวิจัย

1.3.1 สามารถเก็บรวบรวมข้อมูลการผลิตต่อวันรวมถึงการเกิดการหยุดการทำงานของสายการผลิต เพื่อให้ง่ายต่อการนำข้อมูลมาวิเคราะห์

1.3.2 สามารถลดขั้นตอนการจดบันทึกในการตรวจเช็คข้อมูลการผลิตและการหยุดทำงานของสายการผลิตได้

1.3.3 สามารถเช็คจำนวนรถที่ผลิตและจำนวนการเกิดการหยุดการทำงานของสายการผลิตย้อนหลังได้

1.4 ขอบเขตของการวิจัย

1.4.1 ใช้โปรแกรม phpMyAdmin ในการทำระบบจัดการฐานข้อมูล

1.4.2 ใช้ภาษาซีชาร์ป (C# Programming Language) ในการพัฒนาโปรแกรมโปรแกรมแสดงผล และควบคุมสถานะการผลิต โดยพัฒนาบนเครื่องมือ WPF (Windows Presentation Foundation)

1.4.3 ใช้รหัสคิวอาร์ (QR-Code) ในการเพิ่มข้อมูลรถยนต์ที่ผลิต

1.5 ข้อจำกัดของการวิจัย

1.5.1 ความเหมาะสมในการออกแบบหน้าจอแสดงผลเพื่อให้ง่ายต่อการมองเห็น

1.5.1 ข้อจำกัดในการสั่งซื้ออุปกรณ์ในการจัดทำชิ้นงานต้นแบบเพื่อจำลองระบบ

1.6 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

1.6.1 เพิ่มประสิทธิภาพการควบคุมการผลิตโดยแสดงสถานะการผลิตต่อวัน

1.6.2 แสดงเวลามาตรฐานและแสดงสถานะการทำงานในสายการผลิต

1.6.3 เพื่อให้ง่ายต่อการเก็บรวบรวมข้อมูล

1.7 แผนการดำเนินการ

ตารางที่ 1.1 แผนการดำเนินการ

ลำดับ ที่	กิจกรรม	สิงหาคม 63				กันยายน 63				ตุลาคม 63				พฤศจิกายน 63			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	ศึกษาปัญหาและรวบรวมข้อมูล																
2	คัดเลือกปัญหาและศึกษาการเขียนภาษาซีชาร์ป																
3	กำหนดปัญหาที่จะแก้ไข ตั้งเป้าหมาย																
4	เขียนโปรแกรมแสดงผลและควบคุมสถานะการผลิต																

ตารางที่ 1.2 แผนการดำเนินงานช่วงที่ 2

ลำดับ ที่	กิจกรรม	ธันวาคม 63				มกราคม 64				กุมภาพันธ์ 64				มีนาคม 64			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	ติดตั้งอุปกรณ์และทำการทดลอง																
2	ปรับปรุงและแก้ไข																
3	วิเคราะห์และสรุปผลการวิจัย																
4	จัดทำสรุปเล่มวิจัย																
5	นำเสนองานวิจัย																

บทที่ 2

วรรณกรรมที่เกี่ยวข้อง

ปริญญานิพนธ์นี้ได้มีการพัฒนาโปรแกรมแสดงผลและควบคุมสถานการณ์ผลิตโดยผู้วิจัยได้ทำการศึกษาแนวคิดและข้อมูลที่เกี่ยวข้องเพื่อนำมาดำเนินงานวิจัยครั้งนี้โดยถูกแบ่งออกตามประเด็นดังต่อไปนี้

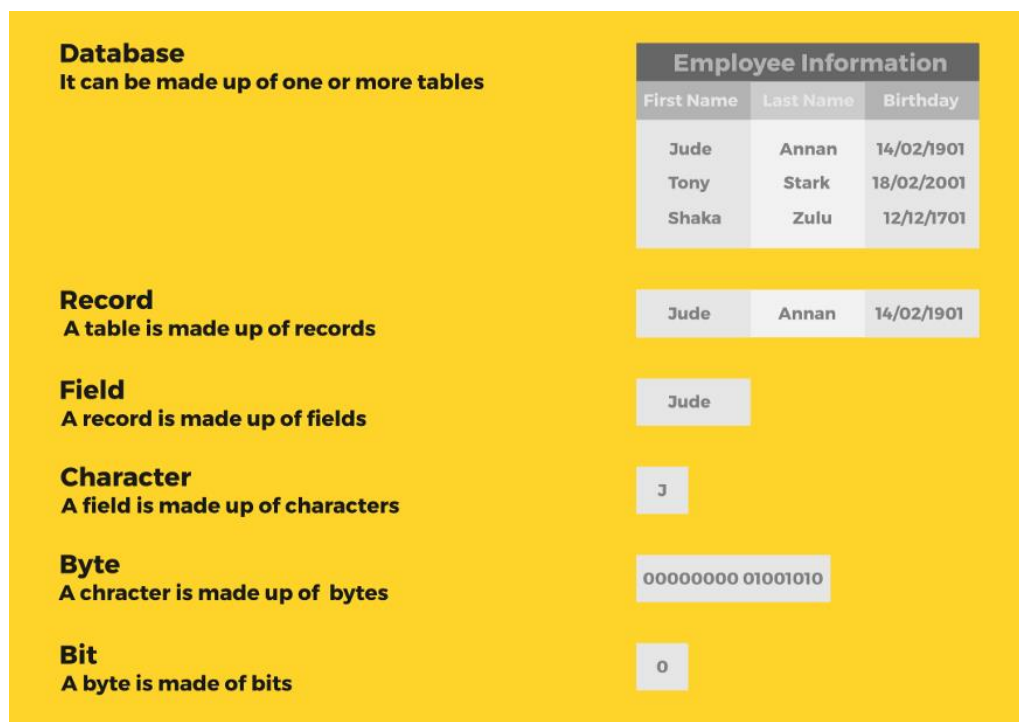
- 2.1 ระบบฐานข้อมูล
- 2.2 โปรแกรมที่ทำหน้าที่ให้บริการ Database
- 2.3 Microsoft Visual Studio
- 2.4 ภาษาซีชาร์ป (C# Programming Language)
- 2.5 แนวคิดเชิงวัตถุ (Object-Oriented Concept)
- 2.6 การควบคุมด้วยการมองเห็น (Visual Control)
- 2.7 ประสิทธิภาพของผู้ใช้งาน และ ส่วนต่อประสานกับผู้ใช้

2.1 ระบบฐานข้อมูล

ระบบฐานข้อมูล (Database System) หมายถึง การรวบรวมข้อมูลหลายข้อมูลมีความสัมพันธ์กัน โดยมีวัตถุประสงค์เพื่อให้สามารถเข้าถึงและจัดการข้อมูลได้ง่าย ลดความซ้ำซ้อนของข้อมูล และทำให้การบำรุงรักษาตัวโปรแกรมง่ายมากขึ้น โดยผ่านระบบการจัดการฐานข้อมูล หรือ เรียกว่า DBMS (Database Management Systems) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล ส่วนประกอบแฟ้มข้อมูล (File) ระเบียบ (Record) และ เขตข้อมูล (Field) และถูกจัดการด้วยระบบเดียวกัน โปรแกรมคอมพิวเตอร์จะเข้าไปดึงข้อมูลที่ ต้องการได้ อย่างรวดเร็ว (GlurGeek,2018)

2.1.1 โครงสร้างข้อมูล (Data Structure)

โครงสร้างของข้อมูลสามารถแบ่งออกเป็นส่วนประกอบต่างๆที่สำคัญโดยสามารถแสดงได้ดังภาพที่ 2.1 และมีการอธิบายในแต่ละส่วนประกอบได้ดังต่อไปนี้



ภาพที่ 2.1 โครงสร้างข้อมูล Data Structure

ที่มา <https://medium.com/we-are-orb/data-hierarchy-101-aef1e62d32c5>

2.1.1.1 บิต (Bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุด

2.1.1.2 ไบต์ (Byte) หมายถึง หน่วยของข้อมูลที่เป็นเลขฐานสองจำนวน 8 หลักหรือที่ใช้แทนข้อมูลที่เป็นตัวอักษร ตัวเลข หรือสัญลักษณ์ต่าง ๆ เพียง 1 ตัว เช่น A B C ก ข ค ง ฯลฯ เป็นต้น

2.1.1.3 เขตข้อมูล (Field) หมายถึง หน่วยของข้อมูลประกอบขึ้นจากตัวอักษรตั้งแต่หนึ่งตัวขึ้นไปมารวมกัน เช่น ชื่อ นามสกุล ที่อยู่ เบอร์โทรศัพท์ เป็นต้น

2.1.1.4 ระเบียบ (Record) หมายถึง การนำข้อมูลหลายๆเขตข้อมูลมารวมกัน เพื่อเกิดเป็นข้อมูลเรื่องใดเรื่องหนึ่งขึ้นมา เช่น ข้อมูลลงเวลาเข้าออกพนักงาน 1 คน จะประกอบด้วย รหัสพนักงาน เวลาเข้า-ออก ชื่อ-นามสกุล และตำแหน่งพนักงาน เป็นต้น

2.1.1.5 แฟ้มข้อมูล (File) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำข้อมูลหลายๆ ระเบียบที่เป็นเรื่องเดียวกันมารวมกัน เช่น แฟ้มข้อมูลนักศึกษา แฟ้มข้อมูลลูกค้า แฟ้มข้อมูลพนักงาน (Crengettummakit, 2018)

2.1.2 องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูล คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องเข้าไว้ด้วยกันอย่างมีระบบ และมีความสัมพันธ์กันที่ชัดเจน โดยฐานข้อมูลจะต้องประกอบไปด้วยองค์ประกอบหลักดังต่อไปนี้

2.1.2.1 แอปพลิเคชันฐานข้อมูล เป็นแอปพลิเคชันที่สร้างไว้ให้ผู้ใช้งานสามารถติดต่อกับฐานข้อมูลได้อย่างสะดวก ซึ่งมีรูปแบบการติดต่อกับฐานข้อมูลแบบเมนูหรือกราฟฟิก โดยผู้ใช้ไม่จำเป็นต้องมีความรู้เกี่ยวกับฐานข้อมูลเลยก็สามารถเรียกใช้งานฐานข้อมูลได้เช่น บริการเงินสด ATM

2.1.2.2 ระบบจัดการฐานข้อมูล คือกลุ่มโปรแกรมหรือซอฟต์แวร์ชนิดหนึ่ง ที่สร้างขึ้นมาเพื่อทำหน้าที่บริหารฐานข้อมูลโดยตรง ให้มีประสิทธิภาพมากที่สุด เป็นเครื่องมือที่ช่วยอำนวยความสะดวกให้ผู้ใช้สามารถเข้าถึงข้อมูลได้ โดยที่ผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างฐานข้อมูล คือ DBMS นี้เป็นตัวกลางในการเชื่อมโยงระหว่างผู้ใช้ และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับระบบฐานข้อมูล ซึ่งหน้าที่ของระบบจัดการฐานข้อมูลสามารถแบ่งได้เป็น 7 ส่วนดังต่อไปนี้

- 1.กำหนดมาตรฐานข้อมูล
- 2.ควบคุมการเข้าถึงข้อมูลแบบต่าง ๆ
- 3.ดูแล-จัดเก็บข้อมูลให้มีความถูกต้องแม่นยำ
- 4.จัดเรียงการสำรอง และฟื้นฟูสภาพแฟ้มข้อมูล
- 5.จัดระเบียบแฟ้มทางกายภาพ (Physical Organization)
- 6.รักษาความปลอดภัยของข้อมูลภายในฐานข้อมูล
- 7.บำรุงรักษาฐานข้อมูลให้เป็นอิสระจากโปรแกรมแอปพลิเคชันอื่นๆ

2.1.2.3 ดาต้าเบสเซิร์ฟเวอร์ คือ คอมพิวเตอร์ที่มีความรวดเร็วในการทำงานสูงที่คอยให้บริการการจัดการฐานข้อมูล

2.1.2.4 ข้อมูล คือ เนื้อหาของข้อมูลที่เราใช้งาน ซึ่งจะถูกเก็บในหน่วยความจำของดาต้าเบสเซิร์ฟเวอร์ โดยจะถูกเรียกมาใช้งานจากระบบจัดการฐานข้อมูล

2.1.2.5 ผู้บริหารฐานข้อมูล ผู้บริหารฐานข้อมูล คือ กลุ่มบุคคลที่ทำหน้าที่ดูแลข้อมูลผ่านระบบจัดการฐานข้อมูล ซึ่งจะควบคุมให้การทำงานเป็นไปอย่างราบรื่น นอกจากนี้ยังทำหน้าที่กำหนดสิทธิ์การใช้งานข้อมูล กำหนดในเรื่องความปลอดภัยของการใช้งาน พร้อมทั้งดูแลดาต้าเบสเซิร์ฟเวอร์ให้ทำงานอย่างปกติด้วย (องค์ประกอบของระบบฐานข้อมูล(Database), 2555)



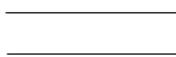



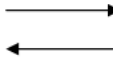
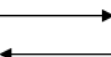
2.1.3 เครื่องมือที่ใช้ในการวิเคราะห์และออกแบบระบบฐานข้อมูล

2.1.3.1 แผนภาพกระแสข้อมูล (Data Flow Diagram: DFD)

แผนภาพกระแสข้อมูล คือ แบบจำลองกระบวนการ (Process Model) ประเภทหนึ่ง มักนำมาใช้ในการวิเคราะห์และออกแบบระบบเชิงโครงสร้าง โดยแสดงความสัมพันธ์ระหว่างกระบวนการกับข้อมูล ซึ่งเป็นการแสดงการเคลื่อนที่ของข้อมูลจากกระบวนการหนึ่งไปยังกระบวนการหนึ่งโดยไม่อิงเทคโนโลยีการจัดเก็บข้อมูล เช่น สื่อบันทึกข้อมูล ฯลฯ

2.1.3.2 สัญลักษณ์ที่ใช้ในการเขียนแผนภาพกระแสข้อมูล

สัญลักษณ์ที่ใช้ในการเขียนแผนภาพกระแสข้อมูลนั้น ประกอบด้วย 4 สัญลักษณ์ที่แสดงถึงการประมวลผล การไหลของข้อมูล ส่วนที่ใช้เก็บข้อมูล และสิ่งที่อยู่นอกระบบ โดยในที่จะใช้สัญลักษณ์ ซึ่งมีรายละเอียด ดังภาพที่ 2.2

DeMarco & Yourdon	Gane & Sarson	ความหมาย
		Process : ขั้นตอนการทำงานภายในระบบ
		Data Store : แหล่งข้อมูลสามารถเป็นได้ทั้งไฟล์ข้อมูลและฐานข้อมูล (File or Database)
		External Agent : บัณฑิตหรือสภาพแวดล้อมที่มีผลกระทบต่อระบบ
		Data Flow : เส้นทางไหลของข้อมูล แสดงทิศทางของข้อมูลจากขั้นตอนการทำงานหนึ่งไปยังอีกขั้นตอนหนึ่ง

ภาพที่ 2.2 สัญลักษณ์ที่ใช้ในการเขียนแผนภาพกระแสข้อมูล

ที่มา <https://sites.google.com/site/krittiyaporn8345/4-2-dfd>

จากภาพที่ 2.2 สัญลักษณ์การเขียนแผนภาพกระแสข้อมูลประกอบด้วยสัญลักษณ์ต่างๆ ซึ่งมีความหมายในตัวเอง ดังนั้นการเขียนแผนภาพกระแสข้อมูลจึงต้องมีกฎเกณฑ์เพื่อแสดงถึงความถูกต้องในการเขียนแผนภาพ ดังนี้

1. กระบวนการ (Process) แสดงการทำงาน หรือการประมวลผลของระบบ โดยที่ตั้งชื่อเพื่อให้สื่อถึงว่า ณ ตำแหน่งหนึ่งๆ ระบบต้องมีกระบวนการทำงานอย่างไร ตามลำดับ

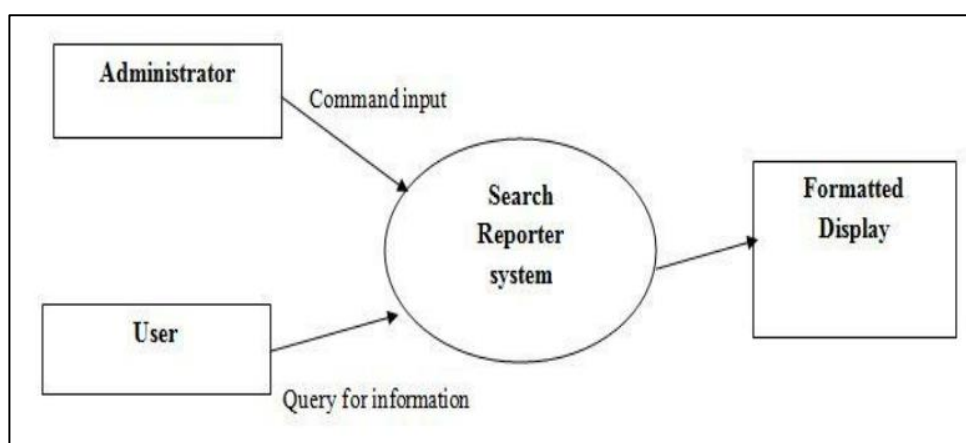
2. การเคลื่อนที่ของข้อมูล (Data Flow) แสดงการเคลื่อนที่ของข้อมูลจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง ของแผนภาพ โดยใช้สัญลักษณ์เส้นและมีลูกศรแสดงทิศทางการเคลื่อนที่ของข้อมูล การเคลื่อนที่ของข้อมูล แสดงเฉพาะการเคลื่อนที่ของข้อมูล เท่านั้น เอกสารบางอย่างถูกจัดอยู่ในกลุ่มของข้อมูล เช่น รายงานต่างๆ เป็นต้น

3. แหล่งเก็บข้อมูล (Data Store) เป็นส่วนที่ใช้แทนชื่อแฟ้มข้อมูลที่เก็บข้อมูล เพราะมีการประมวลผลหลายแบบที่จะต้องมีการเก็บข้อมูลไว้เพื่อที่จะได้นำไปใช้ได้ภายหลัง ซึ่งแหล่งเก็บข้อมูลจะต้องมีทั้งข้อมูลเข้าและข้อมูลออก เป็นต้น

4. สิ่งที่อยู่ภายนอก (External Entity) แสดงสิ่งที่อยู่ภายนอกที่ติดต่อกับระบบ โดยอ้างอิงจากขอบเขตของระบบ (System Boundary) หน่วยภายนอกของระบบ ได้แก่ บุคคล กลุ่มคน หน่วยงาน ซึ่งมีหน้าที่หลักคือ ส่งข้อมูลเข้า และ/หรือ รับข้อมูลที่ออกจากระบบที่กำลังศึกษา

2.1.3.3 คอนเท็กซ์ไดอะแกรม (Context Diagram: Level 0)

Context Diagram: Level 0 คือแผนภาพที่อยู่บนสุด แสดงภาพรวมของระบบทั้งหมด ซึ่งประกอบด้วย กระบวนการเพียงกระบวนการเดียว (อยู่ตรงกลางของภาพ) นั่นคือระบบที่กำลังศึกษา ณ ปัจจุบัน บุคคล/ระบบภายนอก และการเคลื่อนที่ของข้อมูลจากภายนอกกระบวนสู่ระบบดังตัวอย่าง

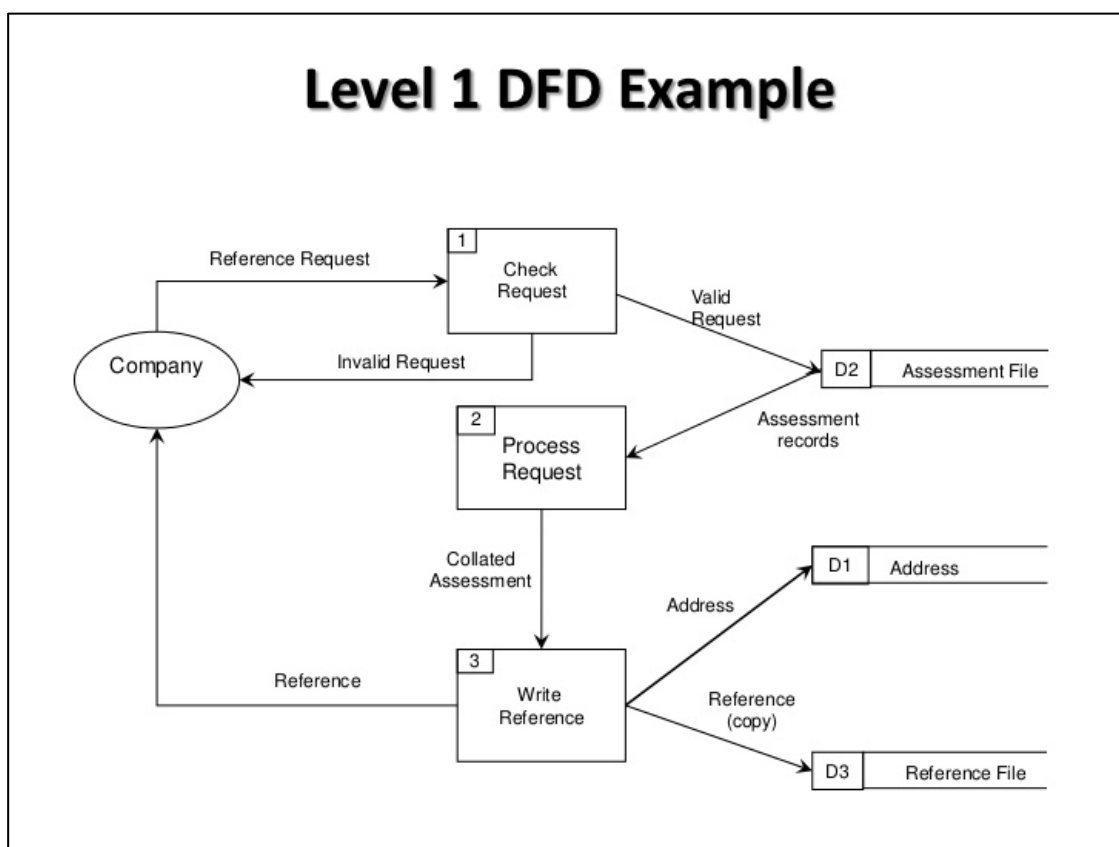


ภาพที่ 2.3 แสดง Context diagram: DFD Level 0

ที่มา <https://www.glurgeek.com/wp-content/uploads/2018/01/Figure-9-Level-0-Data-Flow-Diagram.png>

2.1.3.4 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD Level 1)

แผนภาพกระแสข้อมูลระดับที่ 1 คือ แผนภาพบริบทซึ่งผ่านการแยกออกเป็นส่วน ๆ โดยที่นักวิเคราะห์ระบบยังคงวาด บุคคล/ระบบภายนอก และการเคลื่อนที่ของข้อมูลจากภายนอก ระบบสู่ระบบไว้แต่จะทำการแยกกระบวนการหลัก (ซึ่งอยู่ตรงกลางภาพในแผนภาพบริบท) ให้ออกเป็นกระบวนการย่อย ๆ (Sub Processes) ในแผนภาพกระแสข้อมูลระดับที่ 1 นักวิเคราะห์ระบบใช้ตัวเลขเพื่อเรียกกระบวนการ โดยเริ่มจาก 1.0 ไปเรื่อย ๆ จนกว่าจะครบทุกกระบวนการ หลักการให้ตัวเลขกับกระบวนการนั้น ไม่มีกฎเกณฑ์แน่นอน แต่นิยมให้ตัวเลขเพิ่มขึ้นตามการเคลื่อนที่ของข้อมูลจากกระบวนการหนึ่งสู่ กระบวนการหนึ่ง หากมีการจัดเก็บหรือค้นคืนข้อมูลจากแฟ้มข้อมูล ให้วาดเส้นเชื่อมต่อระหว่างแฟ้มข้อมูลกับกระบวนการที่เกี่ยวข้อง

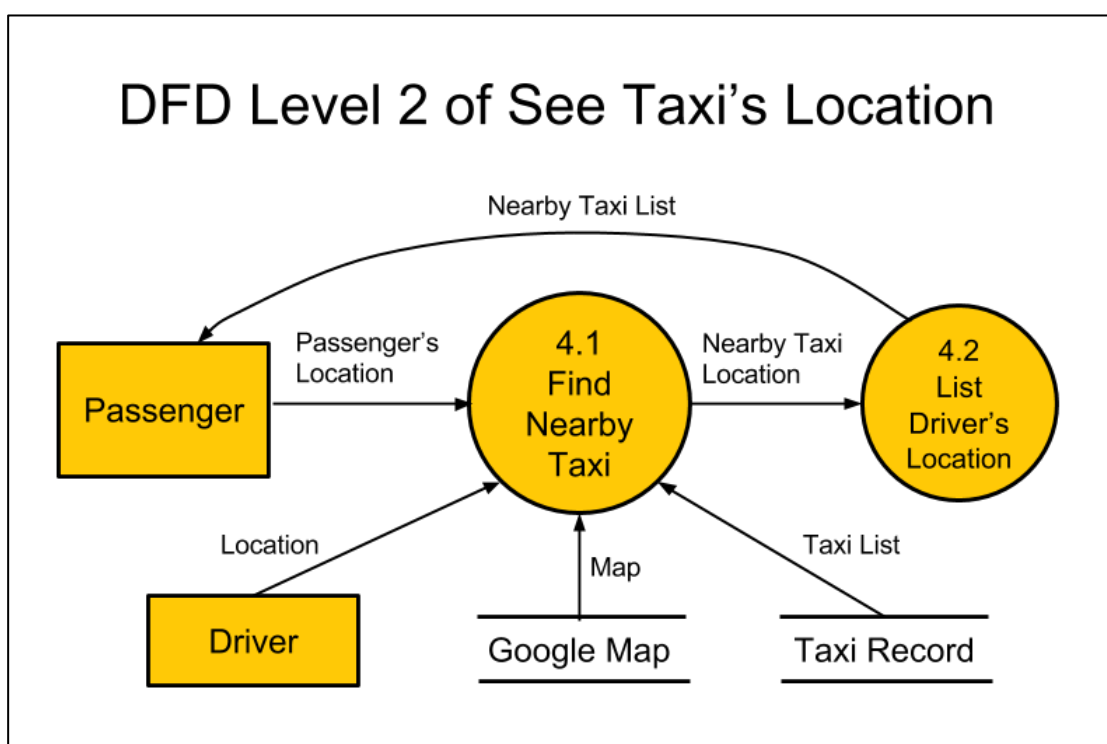


ภาพที่ 2.4 แสดง Data Flow Diagram Level 1

ที่มา <https://www.glurgeek.com/wp-content/uploads/2018/01/btec-national-in-ict-unit-3-data-flow-diagrams-introduction-18-728.jpg>

2.1.3.5 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD Level 2)

แผนภาพกระแสข้อมูลระดับที่ 2 คือ แผนภาพกระแสข้อมูลระดับที่ 1 ซึ่งผ่านการแยกออกเป็นส่วน ๆ โดยนักวิเคราะห์ระบบจะเริ่มจากการเลือกกระบวนการที่อยู่ในแผนภาพกระแสข้อมูลระดับที่ 1 ทีละ 1 กระบวนการ เช่น เลือกกระบวนการหมายเลข 1.0 หลังจากเลือกกระบวนการที่ต้องการศึกษา(1.0) แล้ว นักวิเคราะห์ระบบยังคงวาด การเคลื่อนที่ของข้อมูลเข้าและ การเคลื่อนที่ของข้อมูลออก ที่มีผลต่อกระบวนการที่เลือก (1.0) และทำการแยกกระบวนการที่เลือก (1.0) ให้ออกเป็นกระบวนการย่อย ๆ ในแผนภาพกระแสข้อมูลระดับที่ 2 นักวิเคราะห์ระบบใช้ตัวเลข เพื่อเรียกกระบวนการ ซึ่งเลขตัวแรกเป็นหมายเลขของกระบวนการที่เลือก และเมื่อได้กระบวนการที่ได้รับการที่ผ่านการแยกกระบวนการ เช่น เลือกกระบวนการหมายเลข 1.0 (ในระดับ DFD-1) ดังนั้น ในแผนภาพกระแสข้อมูลระดับที่ 2 กระบวนการย่อยที่อยู่ภายในกระบวนการหมายเลข 1.0 ได้แก่ 1.1 1.2 1.3... ตามลำดับ





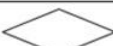






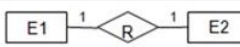
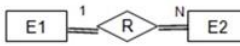
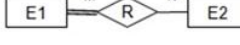
ภาพที่ 2.5 แสดง Data Flow Diagram Level 2

ที่มา <https://www.glurgeek.com/wp-content/uploads/2018/01/DFD-Level-2-of-See-Taxi%E2%80%99s-Location.png>

2.1.3 การออกแบบฐานข้อมูลด้วย ER-Diagram

แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล (E-R Diagram) หมายถึง แผนภาพที่ใช้เป็นเครื่องมือสำหรับจำลองข้อมูล ซึ่งจะประกอบไปด้วย Entity (แทนกลุ่มของข้อมูลที่เป็นเรื่องเดียวกัน / เกี่ยวข้องกัน) และความสัมพันธ์ระหว่างข้อมูล (Relationship) ที่เกิดขึ้นทั้งหมดในระบบ

E-R Diagram หรือ Entity Relationship Diagram จะแสดงชนิดของความสัมพันธ์ว่าเป็น ชนิด หนึ่งต่อหนึ่ง (One to One), หนึ่งต่อหลายสิ่ง (One to Many), หรือ หลายสิ่งต่อหลายสิ่ง (Many to Many) (การสร้างโมเดลความสัมพันธ์ระหว่างข้อมูล: ER-DIAGRAM, n.d.)

สัญลักษณ์ของ Chen	ความหมาย
	เอนติตี้แบบปกติ (Regular Entity)
	เอนติตี้แบบอ่อน (Weak Entity)
	ความสัมพันธ์ (Relationship)
	ความสัมพันธ์แบบเชิงปรากฏ (Identifying Relationship)
	แอททริบิวต์แบบปกติ (Simple Attribute หรือ Atomic Attribute)
	แอททริบิวต์แบบมีหลายค่า (Multivalued Attribute)
	แอททริบิวต์ที่แปลงค่ามา (Derived Attribute)
	แอททริบิวต์ผสม (Composite Attribute)
	แอททริบิวต์ที่ทำหน้าที่เป็นคีย์หลัก (Key Attribute)
	ความสัมพันธ์ของข้อมูลระหว่างเอนติตี้ E1 กับ E2 แบบ 1 : 1 แบบ Partial Participation
	ความสัมพันธ์ของข้อมูลระหว่างเอนติตี้ E1 กับ E2 แบบ 1 : N แบบ Total Participation
	ความสัมพันธ์ของข้อมูลระหว่างเอนติตี้ E1 กับ E2 แบบ M : N โดย E1 เป็น Total Participation และ E2 เป็น Partial Participation

ภาพที่ 2.6 สัญลักษณ์ E-R Model

ที่มา <http://202.29.239.50/temp/2560/class/3-5.php>

2.1.3.1 องค์ประกอบของ ER-Diagram

2.1.3.1.1 เอนทิตี (Entity) คือสิ่งของหรือวัตถุที่เราสนใจ ซึ่งอาจจับต้องได้และเป็นได้ทั้งนามธรรม โดยทั่วไปเอนทิตีจะมีลักษณะที่แยกออกจากกันไป เช่น เอนทิตีพนักงาน จะแยกออกเป็นของพนักงานเลย เอนทิตีเงินเดือนของพนักงานคนหนึ่งก็อาจเป็นเอนทิตีหนึ่งในระบบของโรงงาน

2.1.3.1.2 แอททริบิวต์ (Attribute) คือคุณสมบัติของวัตถุหรือสิ่งของที่เราสนใจ โดยอธิบายรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับลักษณะของเอนทิตี โดยคุณสมบัตินี้มีอยู่ในทุกเอนทิตี เช่น ชื่อ นามสกุล ที่อยู่ แผนก เป็น Attribute ของเอนทิตีพนักงาน

2.1.3.1.3 ความสัมพันธ์ (Relationship) เอนทิตีแต่จะต้องมีความสัมพันธ์ร่วมกัน โดยจะมีชื่อแสดงความสัมพันธ์ร่วมกันซึ่งจะใช้รูปภาพสัญลักษณ์สี่เหลี่ยมรูปว่าวแสดงความสัมพันธ์ระหว่างเอนทิตีและระบุชื่อความสัมพันธ์ลงในสี่เหลี่ยม

2.1.3.2 ระดับชั้นของความสัมพันธ์ (Relationships Degree)

1. ความสัมพันธ์เอนทิตีเดียว (Unary Relationships) หมายถึง เอนทิตีหนึ่ง ๆ จะมีความสัมพันธ์กับตัวมันเอง
2. ความสัมพันธ์สองเอนทิตี (Binary Relationships) หมายถึง เอนทิตีสองเอนทิตีจะมีความสัมพันธ์กัน
3. ความสัมพันธ์สามเอนทิตี (Ternary Relationships) หมายถึง เอนทิตีสองเอนทิตีจะมีความสัมพันธ์กัน

2.3.3.3 การระบุตำแหน่งความสัมพันธ์ระหว่างเอนทิตี (Connectivity)

การระบุตำแหน่งความสัมพันธ์ระหว่างเอนทิตี (Connectivity) ว่าเป็นแบบหนึ่งต่อหนึ่ง (One to One Relationships) , แบบหนึ่งต่อกลุ่ม (One to Many Relationships) หรือ แบบกลุ่มต่อกลุ่ม (Many to Many Relationships) นั้นจะใช้ Connectivity เพื่อระบุตำแหน่ง 1, M หรือ N ไว้ข้างใดของเอนทิตีดังภาพที่ 2.7 – 2.9



ภาพที่ 2.7 ภาพตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง

จากภาพที่ 2.7 เป็นการแสดงความสัมพันธ์ของข้อมูลของเอนทิตีหนึ่งว่ามีความสัมพันธ์กับข้อมูลอย่างมากหนึ่งข้อมูลกับอีกเอนทิตีหนึ่งในลักษณะที่เป็นหนึ่งต่อหนึ่ง



ภาพที่ 2.8 ภาพตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม

จากภาพที่ 2.8 เป็นการแสดงความสัมพันธ์ของข้อมูลของเอนทิตีหนึ่งว่ามีความสัมพันธ์กับข้อมูลหลายข้อมูลกับอีกเอนทิตีหนึ่ง



ภาพที่ 2.9 ภาพตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม

จากภาพที่ 2.9 เป็นการแสดงความสัมพันธ์ของข้อมูลของสองเอนทิตีในลักษณะแบบกลุ่มต่อกลุ่ม

2.2 โปรแกรมที่ทำหน้าที่ให้บริการ Database

2.2.1 มายเอสคิวแอล (MySQL)

คือ โปรแกรมระบบจัดการฐานข้อมูล ที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่นอย่างบูรณาการ เพื่อให้ได้ระบบงานที่รองรับ ความต้องการของผู้ใช้ เช่นทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) เช่น ภาษา php ภาษา asp.net หรือภาษาเจเอสพี เป็นต้น หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น ภาษาวิซวลเบสิกดอทเน็ต ภาษาจาวา หรือภาษาซีชาร์ป เป็นต้น โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลโอเพนซอร์ส (Open Source) ที่ถูกนำไปใช้งานมากที่สุด Mysql จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS : Relational Database Management System) ซึ่งเป็นที่นิยมใช้กันมากในปัจจุบัน (ITGenius, 2014)

2.2.1.1 ชนิดของข้อมูลที่สนับสนุน

ชนิดข้อมูลที่ MySQL สนับสนุนแบ่งเป็น 3 ประเภท ดังนี้

2.2.1.1.1 ชนิดข้อมูลที่เป็นตัวเลข

1. BIT
2. TINYINT
3. SMALLINT
4. MEDIUMINT
5. INT
6. BIGINT
7. Float

2.2.1.1.2 ชนิดข้อมูลที่เกี่ยวข้องกับวันที่และเวลา

1. DATETIME
2. DATE
3. TIMESTAMP
4. TIME
5. YEAR

2.2.1.1.3 ชนิดข้อมูลที่เกี่ยวข้องกับตัวอักษร

1. DATETIME
2. DATE
3. TIMESTAMP
4. TIME
5. YEAR

2.2.1.2 คำสั่งที่ใช้สำหรับการจัดการฐานข้อมูล

ในระบบฐานข้อมูล การบันทึกข้อมูล การปรับปรุงข้อมูลและการลบข้อมูลถือเป็นสิ่งสำคัญ ในภาษา SQL โดยคำสั่งที่ใช้ในการจัดการข้อมูลมี 4 คำสั่ง ดังตารางที่ 2.1

ตารางที่ 2.1 คำสั่งที่ใช้สำหรับการจัดการฐานข้อมูล

คำสั่ง	ความหมาย
SELECT	เรียกค้นข้อมูลในตาราง
INSERT	เพิ่มแถวข้อมูลลงไปในตาราง
UPDATE	ปรับปรุงแถวข้อมูลในตาราง
DELETE	ลบแถวข้อมูลในตาราง

2.2.1.2.1 ส่วนเชื่อมต่อกับภาษาการพัฒนารอื่น (DATABASE CONNECTOR)

ส่วนติดต่อ (interface) เพื่อเชื่อมต่อกับภาษาในการพัฒนา อื่นๆ เพื่อให้เข้าถึงฟังก์ชันการทำงานกับฐานข้อมูล MySQL ได้เช่น ODBC (Open Database Connector) อันเป็นมาตรฐานกลางที่กำหนดมาเพื่อให้ใช้เป็นสะพานในการเชื่อมต่อกับโปรแกรมโดยมีภาษาที่สนับสนุนดังนี้

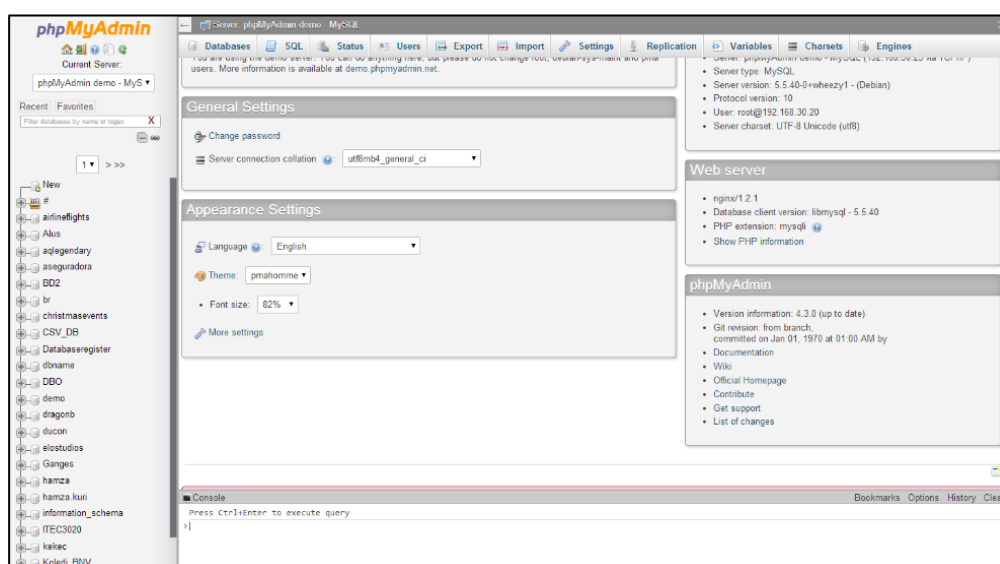
1. DBI สำหรับการเชื่อมต่อกับ ภาษา perl
2. Ruby สำหรับการเชื่อมต่อกับ ภาษา ruby
3. Python สำหรับการเชื่อมต่อกับภาษา python
4. .NET สำหรับการเชื่อมกับภาษา .NET framework
5. MySQL++ สำหรับการเชื่อมต่อกับภาษา C++
6. Ch สำหรับการเชื่อมต่อกับ Ch (C/C++ interpreter)
7. PHP สำหรับการเชื่อมต่อกับภาษาPHP

2.2.2 ฟีเอชพีมายแอดมิน (phpMyAdmin)

phpMyAdmin คือโปรแกรมที่ถูกพัฒนาโดยใช้ภาษา PHP เพื่อใช้ในการบริหารจัดการฐานข้อมูล Mysql แทนการคีย์คำสั่ง เนื่องจากถ้าเราจะใช้ฐานข้อมูลที่เป็น MySQL บางครั้งจะมีความลำบากและยุ่งยากในการใช้งาน ดังนั้นจึงมีเครื่องมือในการจัดการฐานข้อมูล MySQL ขึ้นมาเพื่อให้สามารถจัดการ ตัวDBMS ที่เป็น MySQL ได้ง่ายและสะดวกยิ่งขึ้น

phpMyAdmin เป็นส่วนต่อประสานที่สร้างโดยภาษาพีเอชพี ซึ่งใช้จัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่ หรือทำการสร้าง TABLE ใหม่ๆ และยังมี function ที่ใช้สำหรับการทดสอบการ query ข้อมูลด้วยภาษา SQL พร้อมกันนั้นยังสามารถทำการ insert delete update หรือแม้กระทั่งใช้ คำสั่งต่างๆ เหมือนกับกับการใช้ภาษา SQL ในการสร้างตารางข้อมูล phpMyAdmin เป็นโปรแกรมประเภท MySQL Client ตัวหนึ่งที่ใช้ในการจัดการข้อมูล MySQL ผ่านweb browser ได้โดยตรง phpMyAdmin ตัวนี้จะทำงานบน Web server เป็น PHP Application ที่ใช้ควบคุมจัดการ MySQL Server ความสามารถของ phpMyAdmin มีดังนี้

1. สร้างและลบ Database
2. สร้างและจัดการ Table เช่น แทรก record, ลบ record, แก้ไข record, ลบ Table, แก้ไข field
3. โหลดเท็กซ์ไฟล์เข้าไปเก็บเป็นข้อมูลในตารางได้
4. หาผลสรุป (Query) ด้วยคำสั่ง SQL



ภาพที่ 2.10 ตัวอย่างโปรแกรม phpMyAdmin

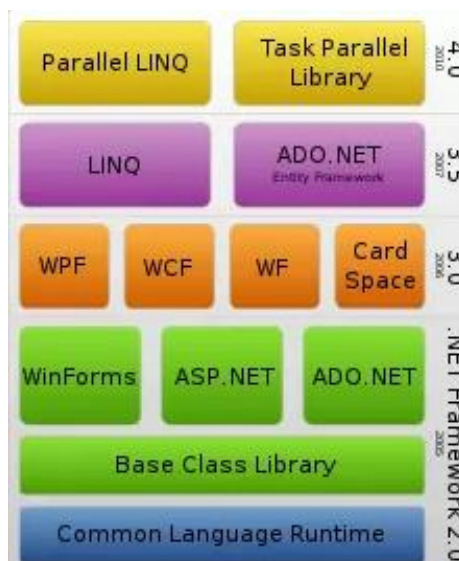
2.3 Microsoft Visual Studio

ในการทำธุรกิจในปัจจุบัน ได้มีการนำเทคโนโลยีต่างๆ เข้ามาช่วยในการทำงาน เนื่องจากสามารถทำงานได้รวดเร็ว และผิดพลาดน้อย สามารถทำงานได้ดีกว่าพนักงานบางคน ดังนั้นนักธุรกิจหรือกิจการส่วนใหญ่จึงได้นำเทคโนโลยีต่างๆ เข้ามาช่วยในการทำงาน รวมถึงระบบของคอมพิวเตอร์ต่างๆ เพื่อช่วยให้อำนวยความสะดวกในการทำงานให้แก่พนักงาน และลดข้อผิดพลาดต่างๆ ลงไป ซึ่งระบบการทำงานต่างๆ นั้น ถูกพัฒนาขึ้นมาโดยนักโปรแกรมเมอร์ ซึ่งผู้พัฒนาจะต้องรู้จักกับภาษาของคอมพิวเตอร์ หรือพูดคุยกับคอมพิวเตอร์ได้ ซึ่งคอมพิวเตอร์นั้นมีหลายภาษา เช่น C++ , C# , Java, VB, VB.NET, PHP, PYTHON, GROOVY และภาษาอื่นๆ อีกมากมาย ซึ่งภาษาแต่ละภาษาก็มีความสามารถและการใช้งานที่แตกต่างกันไป ขึ้นอยู่กับระบบที่ต้องการพัฒนา ในปัจจุบันได้มีเครื่องมือหรือโปรแกรมที่เข้ามาช่วยให้ผู้พัฒนาไม่จำเป็นต้องเขียนภาษาพูดคุยกับคอมพิวเตอร์เองทั้งหมด แต่โปรแกรมจะช่วยพูดคุยกับคอมพิวเตอร์ได้ในระดับหนึ่งแล้ว ซึ่งช่วยอำนวยความสะดวกและลดเวลาการทำงานให้แก่ผู้พัฒนาได้เป็นอย่างมาก ซึ่งโปรแกรมเหล่านี้ก็มีเป็นจำนวนมากเช่นกัน ขึ้นอยู่กับว่าจะใช้ภาษาใดเขียน และโปรแกรมใดที่เหมาะสมกับภาษานั้น ซึ่งจะขอแนะนำโปรแกรมอีกตัวหนึ่งที่ชื่อว่า Visual Studio

Visual Studio (วิซวลสตูดิโอ) คือ โปรแกรมตัวหนึ่งที่เป็นเครื่องมือที่ช่วยพัฒนาซอฟต์แวร์และระบบต่างๆ ซึ่งสามารถติดต่อสื่อสารพูดคุยกับคอมพิวเตอร์ได้ในระดับหนึ่งแล้ว แต่ยังไม่สามารถพัฒนาเป็นระบบเองได้ เหมาะสมสำหรับภาษา VB และ VB.NET เนื่องจากไมโครซอฟต์ได้พัฒนาโปรแกรมและภาษาขึ้นมาควบคู่กันเพื่อให้ใช้งานได้ซึ่งกันและกัน ซึ่งนักโปรแกรมเมอร์จะนำเครื่องมือมาใช้ในการพัฒนาต่อยอดให้เกิดเป็นระบบต่างๆ หรือเป็นเว็บไซต์ และแอปพลิเคชันต่างๆ ดังนั้นวิซวลสตูดิโอจึงเป็นโปรแกรมตัวหนึ่งที่เป็นเครื่องมือที่ช่วยพัฒนาซอฟต์แวร์และระบบต่างๆ ซึ่งโปรแกรมได้มีการติดต่อสื่อสารกับคอมพิวเตอร์ในระดับหนึ่งแล้ว แต่ไม่สามารถพัฒนาเป็นระบบได้ด้วยตนเอง นักพัฒนานำเครื่องมือของโปรแกรมมาใช้ในการพัฒนาต่อให้เกิดเป็นซอฟต์แวร์หรือระบบต่างๆ เพื่อช่วยอำนวยความสะดวก และลดเวลาการทำงานและข้อผิดพลาดได้เป็นอย่างมาก (Visual Studio วิซวลสตูดิโอ คืออะไร, 2559)

2.3.1 ดอตเน็ตเฟรมเวิร์ค (.NET Framework)

ดอตเน็ตเฟรมเวิร์ค (.NET Framework) เป็นแพลตฟอร์มสำหรับพัฒนาซอฟต์แวร์ที่รองรับภาษาดอตเน็ตมากกว่า 40 ภาษา ซึ่งมี Library เป็นจำนวนมากสำหรับการเขียนโปรแกรม รวมถึงการบริหารการดำเนินการของโปรแกรมบน .NET Framework โดย Library นั้นได้รับการพัฒนาแอปพลิเคชัน โดย ดอตเน็ตเฟรมเวิร์ค (.NET Framework) มีส่วนประกอบ ภายในแบ่งออกเป็น 3 ชั้นใหญ่ๆ คือ



ภาพที่ 2.11 ดอตเน็ตเฟรมเวิร์ค (.NET Framework)

ที่มา <https://notebookspec.com/web/88056-net-framework>

1. Programming Language เป็นรูปแบบของภาษาที่ทาง Microsoft ได้เปิดตัว โดยภาษาหลักๆ ที่จะใช้ในการพัฒนามีทั้งหมด 3 ภาษา คือ ภาษาซีชาร์ป ภาษาวิชวลเบสิกดอท เป็นภาษาที่พัฒนามาจาก Visual Basic ภาษาเจเอส สคริป ดอตเน็ต เป็นภาษาที่พัฒนามาจาก JScript ซึ่งเป็น JavaScript ในเวอร์ชันของ Microsoft

2. Base Classes Library นั้นเปรียบเสมือน ชุดคำสั่งสำเร็จรูปย่อย ๆ ที่ต้องใช้งานอยู่เป็นประจำ ดังนั้นจึงมีผู้คิดค้นเครื่องอำนวยความสะดวกในการเขียนโปรแกรม ซึ่ง Library ใน ภาษาต่างๆ ซึ่งภายในระบบ .NET จะสร้างสิ่งที่เรียกว่าเป็น Library พื้นฐานขึ้น ทำให้ไม่ว่าจะใช้ภาษาใดในการพัฒนาโปรแกรม ก็สามารถที่จะเรียกใช้ Library ที่เป็นตัวเดียวกันได้หมด

3. Common Language Runtime (CLR) มีหน้าที่ทำให้ โปรแกรม ที่เขียนขึ้นมาด้วยภาษาต่างๆ กัน กลายเป็นภาษารูปแบบมาตรฐานเดียวกันทั้งหมดเรียกภาษาที่ ว่านี้ว่า Intermediate language (IL) ซึ่งเมื่อต้องการที่จะรัน โปรแกรมใด CLR ที่ว่านี้จะตรวจสอบเครื่องที่รันว่ามีสภาวะแวดล้อมการทำงาน เช่นใดหลังจากนั้นก็คอมไพล์เป็นโปรแกรมที่เหมาะสมต่อการทำงาน ของเครื่องนั้น ทำให้เราสามารถใช้งานโปรแกรมต่างๆ ได้อย่างมีประสิทธิภาพสูงสุด ในแต่ละเครื่อง (Laem, 2011)

2.4. ภาษาซีชาร์ป (C# Programming Language)

ภาษาซีชาร์ป (C# Programming Language) เป็นภาษาเขียนโปรแกรมแบบ multi-paradigm ซึ่งมีคุณสมบัติในการเขียนแบบฟังก์ชัน การเขียนทั่วไป และการเขียนโปรแกรมแบบออบเจ็ค ถูกพัฒนาโดย Microsoft ภายใต้ .NET Framework โดยในการพัฒนา ภาษาซีชาร์ปนี้ มีความตั้งใจให้มันเขียนง่าย ทันสมัย เป็นโปรแกรมเพื่อวัตถุประสงค์ทั่วไป

ในการพัฒนาของภาษาซีชาร์ป นั้นมีความตั้งใจให้เป็นภาษาที่ง่าย ทันสมัย สนับสนุนการเขียนโปรแกรมเพื่อวัตถุประสงค์ทั่วไปและการเขียนโปรแกรมเชิงวัตถุ การพัฒนาของภาษานั้นมีการสนับสนุนสำหรับหลักการของวิศวกรรมซอฟต์แวร์ (Software Engineering) เช่น การตรวจสอบประเภทข้อมูลที่เข้มงวด การตรวจสอบขอบเขตของอาร์เรย์ หรือการพยายามใช้ตัวแปรที่ไม่ได้กำหนดค่า หรือการกำจัดการขยยะอัตโนมัติ ความแข็งแรง ความทนทาน และคุณภาพของโปรแกรม และนอกจากนี้ภาษาซีชาร์ป ได้รวบรวมข้อดีของภาษาต่างๆ เข้าไว้ด้วยกัน ไม่ว่าจะเป็น ภาษาจาวา ภาษาซี และ ภาษาซีพลัสพลัส (ภาษา C#, 2016)

2.4.1 โครงสร้างโปรแกรมภาษาซีชาร์ป

โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐานจะประกอบด้วยส่วนของโปรแกรมหลักแต่จะไม่มีส่วนของโปรแกรมน้อย (subroutine) โดยแสดงดังภาพที่ 2.12-2.13

```
namespace (1)
{
    class (2)
    {
        static void Main ( )
        {
            (3)
        }
    }
}
```

ภาพที่ 2.12 โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน

ที่มา <https://sites.google.com/site/programming42/phas-a-c>

จากภาพที่ 2.12 แสดงโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน โดยมีรายละเอียดดังนี้

1. หมายเลข (1) เป็นการระบุชื่อของ namespace ซึ่งใช้ในการกำหนดขอบเขตให้กับคลาสต่างๆ รวมถึงใช้ในการจัดโครงสร้างของโปรแกรมขนาดใหญ่ให้เป็นสัดส่วนอีกด้วย โดยเฉพาะอย่างยิ่งในการเขียนโปรแกรมคอมพิวเตอร์ที่ซับซ้อนโดยมีผู้เขียนโปรแกรมหลายคน นอกจากนี้ การกำหนด namespace ยังช่วยป้องกันปัญหาการตั้งชื่อคลาสหรือค่าคงที่อื่นๆ ซ้ำกันได้
2. หมายเลข (2) เป็นการระบุชื่อของ class
3. หมายเลข (3) เป็นการระบุพื้นที่สำหรับคำสั่งต่างๆ ที่ผู้เขียนโปรแกรมต้องการให้คอมพิวเตอร์ปฏิบัติตาม

```
namespace HelloApp
{
    class HelloC#
    {
        static void Main ( )
        {
            System.Console.WriteLine ( "Hello C#" );
            System.Console.ReadLine ( );
        }
    }
}
```

ภาพที่ 2.13 ตัวอย่างแสดงข้อความ Hello C# ออกทางจอภาพ

ที่มา <https://sites.google.com/site/programming42/phase-c>

จากภาพที่ 2.13 แสดงตัวอย่าง โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน ซึ่งจะแสดงข้อความ Hello C# ออกทางจอภาพ และจากนั้นรอนจนกว่าผู้ใช้งานจะกด Enter แล้วจบการทำงาน

2.5 แนวคิดเชิงวัตถุ (Object-Oriented Concept)

2.5.1 แนวคิดเชิงวัตถุ (Object-Oriented)

มุ่งเน้นสิ่งต่าง ๆ ที่ใกล้เคียงกับโลกแห่งความจริงในลักษณะรูปธรรมโดยจะมองระบบเป็นกลุ่มของวัตถุ (Object) ที่มีปฏิสัมพันธ์กันด้วยการนำข้อมูลและฟังก์ชันการทำงานรวมเข้าด้วยกันในวัตถุ ทำให้ข้อมูลที่เป็นวัตถุนั้นสามารถอธิบายคุณสมบัติ รวมทั้งฟังก์ชันการทำงานในตัวเองได้ส่วนวิธีการติดต่อกันระหว่างวัตถุจะทำการติดต่อผ่านอินเทอร์เฟซ (Interface) ที่กำหนดไว้ซึ่งแตกต่างกับแนวความคิดแบบดั้งเดิมที่ข้อมูลจะแยกออกจากฟังก์ชัน จึงทำให้ไม่สามารถอธิบายคุณสมบัติของตัวเราได้โลกแห่งวัตถุคือ สิ่งที่เราเผชิญอยู่ในโลกแห่งความจริงในชีวิตประจำวัน สิ่งที่เราเห็นก่อให้เกิดรายละเอียดต่าง ๆ มากมายในวัตถุสิ่ง ๆ นั้น ซึ่งเรียกว่า Abstraction

2.5.2 ออบเจกต์ (Object)

ออบเจกต์ (Object) คือสิ่งที่เราสนใจอาจจะเป็นสิ่งที่จับต้องได้เช่น ลูกค้า สินค้าหรืออาจจะเป็นสิ่งที่จับต้องไม่ได้เช่น หน่วยงานต่าง ๆ ในบริษัท เป็นต้น โดยที่ออบเจกต์ต่าง ๆ สามารถติดต่อสื่อสารกันได้ด้วยการส่งเมสเสจ (Message) ถึงกัน ตัวอย่างเช่น มีลูกค้าต้องการเปิดบัญชีใหม่กับธนาคาร จะเห็นได้ว่าลูกค้าและธนาคาร เป็นออบเจกต์ที่เรากำลังสนใจ โดยลูกค้าจะส่งเมสเสจไปยังธนาคารเพื่อขอเปิดบัญชีแล้วออบเจกต์ธนาคารก็จะส่งเมสเสจกลับไปยังออบเจกต์ลูกค้าโดยส่งหมายเลขบัญชีกลับไปให้

2.5.3 เอนแคปซูเลชัน (Encapsulation)

เอนแคปซูเลชัน คือการรวม attribute และ methods ไว้ด้วยกันเป็นหนึ่งออบเจกต์โดยจะทำหน้าที่ป้องกันมิให้ออบเจกต์อื่นที่อยู่ภายนอกเข้าถึงออบเจกต์หนึ่ง ๆ ได้อย่างอิสระจะมีเฉพาะ methods ที่อยู่ในออบเจกต์เท่านั้นที่สามารถติดต่อกับ attribute ที่อยู่ในออบเจกต์เดียวกันได้เรียกว่าการ Encapsulation มีคุณสมบัติของ Information hiding โดย Information hiding คือการจำกัดการมองเห็นข้อมูลภายในออบเจกต์เช่น การกำหนดคุณลักษณะเป็น Public เพื่อให้สามารถเชื่อมต่อกับออบเจกต์ภายนอกได้และกำหนดเป็น Private เพื่อไม่ให้ออบเจกต์ภายนอกเชื่อมต่อได้นอกจากตัวมันเอง

2.5.4 คลาส (Class)

คลาส (Class) คือกลุ่มของออบเจกต์ที่โครงสร้างพื้นฐานและพฤติกรรมเดียวกันดังนั้นออบเจกต์ที่มีคุณสมบัติลักษณะเดียวกันก็จะรวมกลุ่มอยู่ในคลาสเดียวกัน จึงสรุปได้ว่าคลาสก็คือต้นแบบข้อมูลที่มีไว้เพื่อสร้างออบเจกต์นั่นเอง ประเภทของคลาสอาจจะมีกำหนดได้หลายลักษณะ ดังนั้นคลาสเป็นสิ่งที่อยู่จริงหรือวัตถุที่มีอยู่จริง เช่น คน บริษัท รถยนต์ เป็นต้น หรือคลาสที่มีอยู่ในลักษณะของเอนติตี้ระดับแนวความคิด (Conceptual Entity) เช่น ตารางการสอน ตารางการบิน สมการ

นอกจากนี้ยังมีคลาสที่กำหนดไว้เพื่อใช้เป็นเครื่องมือเช่น อาร์เรย์ (Array), ไบนารีทรี(Binary Tree) เป็นต้น โดยองค์ประกอบของคลาสสามารถแบ่งออกได้ 3 ส่วนคือ

2.5.4.1 ชื่อคลาส (Class name) โดยทุก ๆ คลาสจะต้องมีการกำหนดชื่อคลาส

2.5.4.2 แอตทริบิวต์ (Attribute) เป็นการระบุคุณสมบัติต่าง ๆ ของคลาส โดยที่แต่ละคลาสอาจจะ มีแอตทริบิวต์หรือไม่มีก็ได้

2.5.4.3 แอตทริบิวต์ (Attribute) เป็นการระบุคุณสมบัติต่าง ๆ ของคลาส โดยที่แต่ละคลาสอาจจะ มีแอตทริบิวต์หรือไม่มีก็ได้

2.5.4.4 โอเปอเรชัน (Operation) หรือเมธอด (Method) เป็นการกำหนดฟังก์ชันหรือขั้นตอนซึ่งจะ ถูกกระทำจากออบเจกต์ที่เป็นสมาชิกของคลาสนั้นๆ

จากการที่คลาสมีคุณสมบัติของการเอนแคปซูเลชัน ทำให้คลาสสามารถแบ่งระดับการเข้าถึงองค์ประกอบต่าง ๆ ภายในคลาสที่กล่าวมาแล้ว โดยจะมีการกำหนดสัญลักษณ์เพื่อแสดงคุณสมบัติดังกล่าวอยู่หน้าองค์ประกอบต่าง ๆ ของคลาส การเอนแคปซูเลชันสามารถแบ่ง ออกได้ 3 ระดับดังนี้

1. ระดับ Public หมายถึงองค์ประกอบที่มีสัญลักษณ์ + อยู่ข้างหน้า จะยอมให้คลาสอื่นเข้าถึง และเรียกใช้งานได้

2. ระดับ Private หมายถึงองค์ประกอบที่มีสัญลักษณ์ อยู่ข้างหน้า จะสามารถเรียกใช้หรือ อ้างถึงได้เพียงคลาสเดียวเท่านั้น และไม่สามารถอ้างถึงหรือเรียกใช้จากคลาสลูกได้

3. ระดับ Protected หมายถึงองค์ประกอบที่มีสัญลักษณ์ + อยู่ข้างหน้า จะสามารถเรียกใช้ หรืออ้างถึงได้เพียงคลาสเดียว แต่อนุญาตให้คลาสลูกสามารถใช้งานได้ด้วย

2.5.5 การสืบทอดคุณสมบัติ (Inheritance)

เป็นวิธีการในการสร้างคลาสใหม่จากคลาสเดิมที่มีอยู่ ทั้งนี้คลาสที่สร้างขึ้นใหม่จะมี วัตถุประสงค์ในการทำงานที่เฉพาะเจาะจงมากยิ่งขึ้น ในที่นี้คลาสที่ถูกถ่ายทอดคุณสมบัติจะถูกเรียกว่าซูเปอร์คลาส (Superclass) และเรียกคลาสที่ได้รับการสืบทอดคุณสมบัติว่าซับคลาส (Subclass) ยกตัวอย่างเช่น สิ่งมีชีวิตนั้นสามารถแบ่งเป็นพืชหรือสัตว์ จึงกล่าวได้ว่า คลาสพืช และ คลาสสัตว์นั้นเป็นประเภทหนึ่งของคลาสสิ่งมีชีวิตนั่นเอง

2.5.6 ความสัมพันธ์ระหว่างคลาส (Class Relationship)

ความสัมพันธ์ระหว่างคลาส (Class Relationship) Association เป็นการแสดงความสัมพันธ์ระหว่างคลาสของออบเจกต์โดยความสัมพันธ์ดังกล่าวใช้เชื่อมออบเจกต์ที่เป็นออบเจกต์อิสระของคลาส 2 คลาส

2.5.7 การพัฒนาระบบเชิงวัตถุ

วิธีการพัฒนาระบบแบบดั้งเดิม ขั้นตอนการวิเคราะห์และออกแบบจะมุ่งไปที่รูปแบบ ของกระบวนการแก้ปัญหา เพื่อจัดการข้อมูลเริ่มต้นให้เป็นผลลัพธ์ที่ต้องการเรียกว่า Structured Approach ดังนั้นในช่วงขั้นตอนของการวิเคราะห์และออกแบบจะสร้างไดอะแกรมการไหลข้อมูล (Data Flow Diagrams) ที่ไม่สัมพันธ์กับการสร้างรูปแบบข้อมูล (data modeling) ไดอะแกรมการไหลข้อมูลจะแยกองค์ประกอบแบบบนลงล่าง (เป็นวิธีการกำหนดลักษณะปัญหาหลัก ๆ ในชั้นบน แล้วแตกปัญหาแต่ละส่วนเป็นปัญหาย่อย ๆ ในชั้นถัดลงมา ทำให้ได้รายละเอียดของปัญหาที่ สมบูรณ์ขึ้นเรื่อย ๆ จนถึงขั้นสุดท้าย) อันเป็นลักษณะโครงสร้างแบบลำดับชั้น(Hierarchy) ส่วน ระเบียบวิธีการสร้างรูปแบบข้อมูลจะเกี่ยวกับการกำหนดออบเจกต์ ความสัมพันธ์ระหว่างออบเจกต์ และการทำงานของออบเจกต์ เพื่อสรุปเป็นไดอะแกรมความสัมพันธ์เชิงออบเจกต์ วิธีการแบบดั้งเดิมนี้จะมีจุดเด่นที่โครงสร้างปัญหาชัดเจนแน่นอน การทำงานในแต่ละจุดมี รายละเอียดเฉพาะส่วนของตัวเอง ไม่ก้าวก่ายหรือยุ่งเกี่ยวกับสายงานที่ไม่ใช่สายงานของตัวเอง แต่ไม่เหมาะสำหรับการใช้งานจริงเพราะโครงสร้างปัญหาส่วนใหญ่ไม่ได้เป็นแบบลำดับชั้น อีกทั้งงาน แต่ละจุดไม่มีความสัมพันธ์ต่อกันทั้งหมด ทำให้การประสานงานเป็นเรื่องยุ่งยากมาก แต่สำหรับแนวคิดเชิงวัตถุจะเน้นการสร้างระบบตามการใช้งานจริง โดยวัตถุต่าง ๆ จะต้อง ประกอบด้วยข้อมูลประจำตัว (data) และวิธีการใช้ข้อมูลนั้น (method) ซึ่งวัตถุต่าง ๆ จะถูกแบ่ง หน้าที่ชัดเจน ไม่ซ้ำซ้อนและมีอิสระต่อกัน (independence) จึงมีการประสานการทำงาน ร่วมกันได้ดี นอกจากนี้ยังมีเหตุผลอื่น ๆ ดังนี้

2.5.7.1 การออกแบบ การเขียนโปรแกรม การทดสอบและการบำรุงรักษาทำได้สะดวกขึ้น เพราะระบบแบ่งหน้าที่การทำงานและข้อมูลที่ใช้เป็น Module ชัดเจน ดังนั้นจึงสามารถจัดการปัญหาเฉพาะจุดที่ต้องการได้ โดยไม่สับสน วุ่นวายในการค้นหาจุดที่ต้องการจากทั้งระบบ โดยเฉพาะระบบที่มีองค์ประกอบมากและซับซ้อน

2.5.7.2 การเปลี่ยนแปลงแต่ละขั้นตอนในการพัฒนาระบบทำได้ราบรื่น เพราะระบบเชิงวัตถุจะวางรูปแบบการทำงานไปในแนวทางเดียวกันทั้งหมด อีกทั้งมีการทำงานอย่างเป็นระบบ ชัดเจน

2.5.7.3 สนับสนุนเทคนิคการโปรแกรมที่ดี นั่นคือมีการพัฒนาเทคโนโลยีทางด้านการติดต่อกับผู้ใช้แบบยูสเซอร์อินเทอร์เฟซ (Graphic User Interface: GUI) และการสร้างต้นแบบของคลาส (class) เพื่อซ่อนวิธีการทำงาน (method) ไว้ภายใน ทำให้การเรียกใช้งานแต่ละครั้งเพียง แคบอกระบบว่าให้ทำอะไรบ้าง โดยไม่ต้องอธิบายทุกครั้งว่าต้องทำอะไร

2.5.7.4 สนับสนุนการนำกลับมาใช้ใหม่ (reusability) เมื่อมีการกำหนดองค์ประกอบของระบบเป็นวัตถุ ซึ่งประกอบด้วยข้อมูล (data) และวิธีการทำงาน (method) ที่ชัดเจนเป็นมาตรฐานทำให้การเรียกใช้ข้อมูลแต่ละครั้ง จึงไม่จำเป็นต้องอธิบายการทำงานซ้ำทุกครั้ง เพียงแค่ส่งเมสเสจ (message) บอกวัตถุนั้นให้ทำอะไรบ้างเท่านั้น เป็นการนำโปรแกรมชุดเดิมมาใช้โดยไม่ต้อง สร้างใหม่

แต่สำหรับแบบ Structured Approach ข้อมูลต่างๆ ไม่ได้กำหนดวิธีการทำงานของมัน ชัดเจน การเรียกใช้งานจึงต้องอธิบายวิธีการทำงานทุกครั้งซึ่งเป็นการสิ้นเปลืองโปรแกรมและระบบ ใหญ่เกินความจำเป็น

2.5.8 แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented Database Model)

แบบจำลองฐานข้อมูลเชิงวัตถุเป็นวิธีการใหม่ที่ใช้ในการออกแบบฐานข้อมูล เพื่อให้สามารถจัดเก็บข้อมูลที่มีลักษณะซับซ้อน (Complex Data Type) ได้เช่น ข้อมูลภาพและเสียง เป็นต้น ซึ่งรองรับกับการออกแบบระบบงานเชิงวัตถุ (Object-Oriented Design) โดยในการออกแบบงานใด ๆ ในลักษณะเชิงวัตถุนี้จะใช้เทคนิคของแบบจำลองเชิงวัตถุ ตั้งแต่การวิเคราะห์ระบบ การออกแบบระบบงาน การออกแบบฐานข้อมูล รวมไปถึงการพัฒนาระบบด้วย ซึ่งสามารถบ่งบอกถึงขั้นตอนการทำงานทั้งหมดภายในระบบได้ตามแผนภาพที่นำมาใช้ในการออกแบบ แบบจำลองฐานข้อมูลเชิงวัตถุที่นำมาทำการศึกษาสำหรับงานวิจัยนี้คือ ยูเอ็มแอล (Unified Modeling Language : UML) ซึ่งเป็นแบบจำลองที่มีการรวบรวมมาจากหลักการของ Booch Rumbaugh และ Jacobson โดยมีการกำหนดเป็นมาตรฐานในการใช้งานจาก OMG (Object Management Group) ตั้งแต่ราวต้นปี ค.ศ. 1997 และเป็นที่ยอมรับและมีการใช้งานกันอย่างแพร่หลายของแบบจำลองเชิงวัตถุ ตั้งแต่การวิเคราะห์ระบบ การออกแบบระบบงาน การออกแบบฐานข้อมูล รวมไปถึงการพัฒนา ระบบด้วย

2.5.9 ยูเอ็มแอล (UML)

ยูเอ็มแอล (UML) ย่อมาจาก Unified Modeling Language เป็นภาษาเพื่อใช้อธิบายโมเดลต่างๆ หรือเป็นภาษาสัญลักษณ์รูปภาพมาตรฐานสำหรับใช้ในการสร้างโมเดลเชิงวัตถุ โดย UML เป็นภาษามาตรฐานสำหรับสร้างแบบพิมพ์เขียว (blueprint) ให้แก่ระบบงาน เราสามารถใช้ UML ในการสร้างมุมมอง กำหนดรายละเอียด สร้างระบบงานและจัดเอกสารอ้างอิงให้แก่ระบบงานได้ เนื่องจาก UML เป็นภาษาที่มีการใช้สัญลักษณ์รูปภาพจึงอาจมีผู้เข้าใจสับสนว่า UML เป็นการสร้างไดอะแกรมหรือเป็นเพียงการใช้สัญลักษณ์เพื่ออธิบายระบบงานเท่านั้น แต่แท้จริงแล้ว UML มีลักษณะของเมต้าโมเดล (meta model) คือเป็นโมเดลที่เอาไว้อธิบายโมเดลอื่น ๆ อีกที การใช้งานภาษา UML นอกจากจะต้องเข้าใจในแนวความคิดเชิงวัตถุแล้วยังจำเป็นต้องมีพื้นฐานความเข้าใจเกี่ยวกับ Visual Modeling ด้วยเช่นกันโมเดลลิง (Modeling) เป็นวิธีการวิเคราะห์ออกแบบ (Analysis and Design) อย่างหนึ่งที่เน้นการใช้งานโมเดลเป็นหลัก ซึ่งโมเดลที่สร้างขึ้นจะสามารถช่วยให้เข้าใจในปัญหาได้ง่ายขึ้น อีกทั้งยังสามารถนำโมเดลมาเป็นเครื่องมือในการสื่อสาร ถ่ายทอดความคิดกับบุคคลอื่นๆ ที่เกี่ยวข้องในโครงการได้ เช่น ลูกค้า นักวิเคราะห์ระบบ นักออกแบบระบบ เป็นต้น ส่วน Visual Modeling คือการใช้สัญลักษณ์รูปภาพในการสร้างโมเดลของระบบที่จะพัฒนาเพื่อประโยชน์ที่คล้ายคลึงกันในการทำความเข้าใจกับความต้องการของลูกค้า การออกแบบระบบที่

เป็นไปได้อย่างชัดเจนขึ้นและการบำรุงรักษาที่ง่ายยิ่งขึ้น โมเดลเกิดขึ้นโดยการ นำเสนอส่วนต่าง ๆ ของระบบแต่เพียงส่วนที่สำคัญโดยไม่คำนึงถึงรายละเอียดที่ไม่สำคัญในการ พัฒนาระบบซอฟต์แวร์ที่ซับซ้อน นักพัฒนาจำเป็นต้องทำความเข้าใจกับมุมมองด้านต่าง ๆ ของ ระบบก่อนทำการพัฒนาจริง โดยการสร้างโมเดลอันเปรียบเสมือนพิมพ์เขียวที่แสดงถึงภาพรวมของ ระบบทั้งหมด โมเดลที่สร้างขึ้นจะต้องมีความสอดคล้องกับความต้องการของผู้ใช้งานระบบเป็นสำคัญ ในส่วนของรายละเอียดต่าง ๆ จะค่อย ๆ ถูกเพิ่มเติมลงไปในตัวโมเดล และในที่สุดโมเดลจะ ถูกนำไปพัฒนาขึ้นเป็นระบบจริง

2.6 การควบคุมด้วยการมองเห็น (Visual Control)

การบริหารโรงงานด้วยหลักการมองเห็น (Visual Factory Management) เป็นระบบที่ใช้สนับสนุนการปรับปรุงผลผลิตภาพทั่วทั้งโรงงานโดยครอบคลุมถึงปัจจัยต่าง ๆ ดังเช่น ความปลอดภัย คุณภาพ การส่งมอบตรงเวลา การสร้างผลกำไร และการสร้างขวัญ-กำลังใจ (Employee Moral) โดยมุ่งแสดงด้วยสัญญาณ แอปสี และสัญลักษณ์ต่าง ๆ ในสถานที่ทำงาน เพื่อให้พนักงานหรือผู้เกี่ยวข้องได้รับทราบและเข้าใจสารสนเทศต่าง ๆ ในเวลาอันรวดเร็ว ซึ่งสอดคล้องกับแนวคิดลีน (Lean) สำหรับการดำเนินการบริหารโรงงานด้วยหลักการมองเห็นจะเริ่มด้วยการจัดทำกิจกรรม 5ส. เพื่อจำแนกปัญหาต่าง ๆ ที่เกิดขึ้นในพื้นที่ทำงานและใช้เป็นสารสนเทศสำหรับป้องกันความสูญเสีย ดังนั้น หลักการ Visual Factory Management จึงเป็นเครื่องมือสนับสนุนการบริหารด้วยการแสดงสารสนเทศต่าง ๆ ที่เกี่ยวข้อง เช่น รายละเอียดของงาน สภาพพื้นที่การทำงาน (Work Area Environment) และประเภทเครื่องจักร/วัสดุที่ใช้ เพื่อให้การดำเนินกิจกรรมการผลิตไปอย่างต่อเนื่องและเกิดความปลอดภัยในขณะทำงาน

2.6.1 Visual Factory Management

2.6.1.1 Visual Display เป็นการแสดงสารสนเทศเพื่อให้พนักงานในฝ่ายงานหรือผู้ปฏิบัติงานในพื้นที่ได้รับทราบ โดยมีการนำเสนอในรูปของแผนภูมิและกราฟ ดังเช่น การใช้กราฟ/แผนภูมิ เพื่อแสดงยอดขายรายเดือน (Monthly Revenues) การแสดงข้อมูลผลการปฏิบัติงาน

2.6.1.2 Visual Control หรือการควบคุมด้วยการมองเห็น เป็นวิธีควบคุมบริหารเพื่อใช้เป็นแนวทางปฏิบัติงานและควบคุมให้การทำงานเป็นไปอย่างถูกต้อง โดยแสดงมาตรฐานเทียบกับสถานะจริงทำให้สามารถระบุความบกพร่องได้ทันทีด้วยการมองเห็น นั้นหมายถึง การนำเสนอข้อมูลที่มีอยู่มาเสนอให้เข้าใจได้ง่ายขึ้นด้วยการแปลงข้อมูลให้อยู่ในรูปแบบของตาราง, ป้าย สติกเกอร์ กระดาน สัญลักษณ์, ภาพ, แผนภาพ เป็นต้น แต่การนำเสนอต้องมีความหมายและสาระดึงดูดให้เกิดความน่าสนใจ เพื่อนำข้อมูลมาใช้ติดตามงานหรือเป็นเครื่องมือช่วยย้ำเตือนเป้าหมายต่าง ๆ ดังเช่น มาตรฐานการผลิต วิธีการทำงาน กำหนดการผลิตในแต่ละวัน หัวข้อการควบคุม การระบุตำแหน่งจัดวางวัสดุ กฎระเบียบและข้อห้ามต่าง ๆ ป้ายแสดงตำแหน่งที่จอดรถ ทำให้ผู้รับผิดชอบทราบความแตกต่างระหว่างเป้าหมายกับผลลัพธ์ที่เกิดขึ้นจริง รวมทั้งลดความสูญเสียเวลาสำหรับการค้นหาและ

ติดตามสารสนเทศ สารสนเทศที่ได้รับจากระบบควบคุมด้วยการมองเห็นยังช่วยให้พนักงานสามารถ ประเมินปัญหาและค้นหาแนวทางแก้ไขได้อย่างรวดเร็ว ดังนั้นจึงมักถูกใช้ประยุกต์กับการไหลของงาน หรือการบริหารพื้นที่ทำงานประจำวันเพื่อเป็นแนวทางสำหรับควบคุมด้วยตนเอง (Self-controlling) และเป็นองค์ประกอบหลักของการดำเนินตามแนวทางของลีนที่มุ่งขจัดความผันแปรที่เกิดขึ้นจาก ปัจจัยของกระบวนการ นั่นคือ เครื่องจักร (Machine), วัสดุ (Material), วิธีการ (Method), แรงงาน (Manpower) รวมทั้งความผันแปรของผลิตผลที่ประกอบด้วย คุณภาพ การส่งมอบ และต้นทุน (Quality, Delivery, Cost) การควบคุมด้วยการมองเห็นจะเกิดประสิทธิภาพและประสิทธิผลในพื้นที่ ทำงานจะต้องได้รับการสนับสนุนด้วยระบบการบริหารจัดการด้วยการมองเห็น ซึ่งเป็นวิธีการบริหารจัดการ ใช้สารสนเทศในสถานที่ทำงานอย่างชัดเจนจนมองเห็นได้ง่ายสำหรับผู้รับผิดชอบเพื่อจำแนกความ ผิดปกติที่เกิดขึ้นได้ทันทีด้วยการแบ่งปันสารสนเทศให้ทุกคนได้รับรู้โดยมีการแจ้งกลับสถานะของการ ดำเนินงานแบบเวลาจริงซึ่งเป็นเสมือนระบบประสาทของโรงงาน โดยมุ่งการติดตามกิจกรรมต่าง ๆ ที่ ดำเนินภายในโรงงานเป็นไปอย่างต่อเนื่อง ดังเช่น การแสดงข้อมูลการเกิดของเสียและปัญหาที่เกิดขึ้น ไว้ในตำแหน่งสูงไม่เกิน 4 ฟุต เพื่อให้ผู้ควบคุมสามารถมองเห็นได้ง่ายเมื่อต้องการติดตามตรวจสอบ และดำเนินการแก้ไขอย่างทันเวลา ดังนั้นหลักการ Visual Displays และ Visual Control จึง สนับสนุนให้การดำเนินกิจกรรมต่าง ๆ เป็นไปอย่างมีประสิทธิภาพโดยมุ่งให้พนักงานได้รับทราบ สถานะปัญหาที่เกิดขึ้นได้อย่างรวดเร็ว (Visual Control คืออะไร, 2015)

TRIM 6		PRODUCTION CONTROL BOARD						
HEADLINING ANTOLIN		SET UP	BREAK DOWN	QUALITY	WAITING PART	WAITING PALLET	WAITING KANBAN	FINAL
STATION		002	016	003	004	000	000	000
1	WATER JET & FINAL	OUT PUT CONTROL				PRODUCTION		
2	FORMING PRESS	WORKING TIME PLAN	0030	Min	OA	32 %		
3	COVER SHEET & CUTTING	LOADING TIME	0037	Min				
4	GLASS FIBER & FLEECE	TARGET	0221	Pcs. <th rowspan="2">OE</th> <td colspan="2">36 %</td>	OE	36 %		
5	GLUE COATING	ACTUAL	0026	Pcs.				
6	PU PICK UP	DIFFERENT	-195	Pcs. <td colspan="4" rowspan="2">0006</td>	0006			
		TAKT TIME (SEC)	0001000					

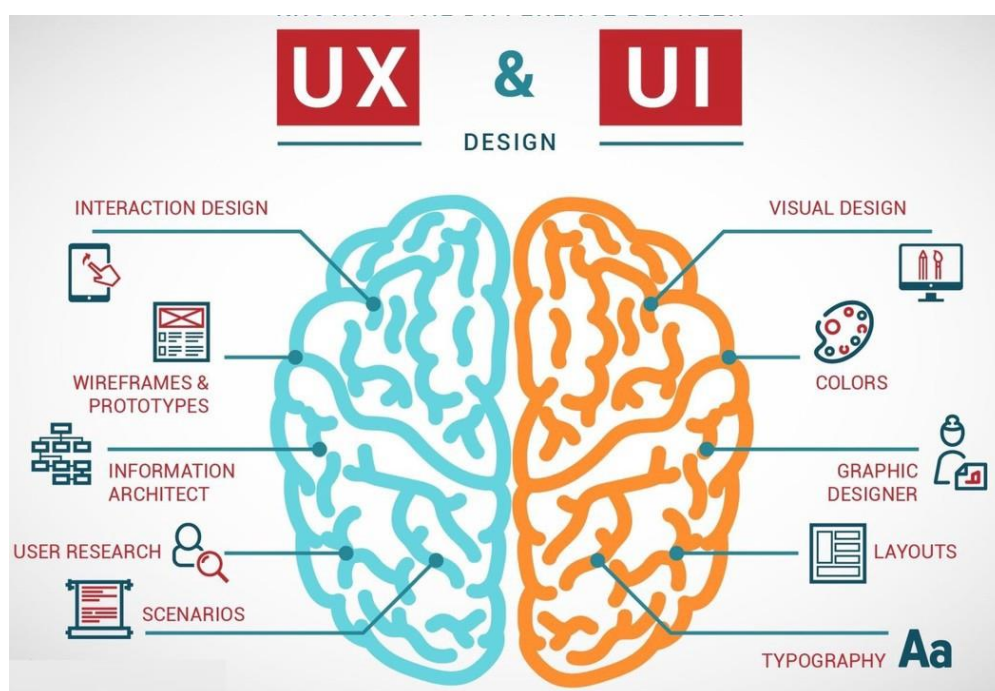
ภาพที่ 2.14 ตัวอย่างแสดงบอร์ดสื่อสารในสายการผลิต

ที่มา <http://interfacecom.thailandpages.com/product/oe-realtime-monitoring-board-28395/>

2.6.1.3 การใช้สัญญาณเสียง (Audio Signals) เพื่อใช้แจ้งเตือนปัญหาที่เกิดขึ้นในโรงงาน หรืออาจเรียกว่า Sound Warning เช่น การเกิดปัญหาเครื่องจักรขัดข้องในสายการผลิต นอกจากนี้ ยังใช้สำหรับการแจ้งเวลาเริ่มต้นและหยุดพักการทำงาน

2.6.1.4 สารสนเทศการมองเห็น (Visual Information) เพื่อใช้ป้องกันความผิดพลาด (Prevent Mistake) ที่อาจเกิดขึ้นจากการปฏิบัติงาน ซึ่งมักแสดงด้วยรหัส/แถบสี (Color Coding) หรือการใช้เครื่องหมายแสดงระดับความปลอดภัย (Safe Range) ดังเช่นการใช้แถบสีแสดงระดับน้ำมันและการใช้ฉลากหรือสติ๊กเกอร์เพื่อจัดแยกประเภทชิ้นงานในสายการประกอบ

2.7 ประสบการณ์ของผู้ใช้งาน และ ส่วนต่อประสานกับผู้ใช้



ภาพที่ 2.15 ประสบการณ์ของผู้ใช้งาน และ ส่วนต่อประสานกับผู้ใช้

ที่มา <https://www.dailytech.in.th/ui>

2.7.1 ประสบการณ์ของผู้ใช้งาน (User Experience) หรือ UX คือ ประสบการณ์ของผู้ใช้งานในด้านความรู้สึกที่ตอบสนองต่อการใช้งานผลิตภัณฑ์ หรือระบบต่าง ๆ ยกตัวอย่างเช่น ความสะดวกสบาย ใช้งานง่าย ความสนุกสนาน จนเกิดเป็นความพึงพอใจสูงสุดหรือเกิดประสบการณ์ที่ดีของผู้ใช้งานนั่นเอง ฉะนั้นในอีกแง่หนึ่ง User experience หรือ UX มีการพัฒนามาจากผลของการปรับปรุง UI เมื่อมีบางอย่างให้ผู้ใช้ได้โต้ตอบกับประสบการณ์ของพวกเขา ไม่ว่าจะเป็นแง่บวก ลบ

หรือเป็นกลาง สามารถเปลี่ยนวิธีที่ผู้ใช้รู้สึกเกี่ยวกับการโต้ตอบเหล่านั้น UX จึงเป็นจุดที่ต้องพยายามศึกษาและทำความเข้าใจว่าผู้ใช้งานต้องการอะไร แบบไหน พอใจไหม กลุ่มเป้าหมายมีใครบ้าง มีอะไรน่าสนใจบ้าง อย่างละเอียด เพื่อให้ตอบโจทย์กับผู้ใช้งานมากที่สุด ฉะนั้นอาจเปรียบได้ว่า UX คือ “ศาสตร์แห่งความพยายามเข้าใจผู้อื่นเพื่อประโยชน์อันสูงสุด”

2.7.2 ส่วนต่อประสานกับผู้ใช้ (User Interface) หรือ UI คือ ส่วนต่อประสานกับผู้ใช้งาน หรือส่วนที่ใช้ในการเชื่อมต่อกับผู้ใช้งาน กล่าวคือ ส่วนที่ให้ผู้ใช้งานสามารถโต้ตอบกับการใช้งานผลิตภัณฑ์ได้ ซึ่งจะมุ่งเน้นไปที่เรื่องของหน้าตา การออกแบบ และการดีไซน์ ยกตัวอย่างเช่น หน้าจอ แพลตฟอร์ม เมนู รูปแบบต่าง ๆ การวางภาพ ขนาดตัวอักษร ปุ่ม แป้นพิมพ์ เสียง หรือแม้แต่แสงไฟ สิ่งสำคัญสำหรับส่วนต่อประสานกับผู้ใช้ ก็คือดีไซน์ที่ดูสะอาด สวยงาม ดึงดูดใจ อีกทั้งต้องเข้าใจง่าย ใช้งานง่าย มีมาตรฐานและเป็นมิตรต่อผู้ใช้งาน นอกจากนี้ยังต้องมีฟังก์ชันที่น่าสนใจ มีภาษาภาพที่ทำให้คนเกิดความรู้สึกอยากใช้งาน และที่สำคัญจะต้องมีความเป็นเอกลักษณ์โดดเด่น แต่ก็ไม่ฉูดฉาดหรือต่างมากเกินไป ฉะนั้นอาจเปรียบได้ว่าส่วนต่อประสานกับผู้ใช้ คือ “ศาสตร์แห่งความสวยงาม” ที่จะมาเติมเต็มให้ประสบการณ์ของผู้ใช้งาน ออกมาเป็นรูปร่างจนเกิดเป็น first impression ที่ดีที่สุดสำหรับผู้ใช้งานนั่นเอง

2.7.3 ความแตกต่างระหว่าง ประสบการณ์ของผู้ใช้งาน และ ส่วนต่อประสานกับผู้ใช้

หากจะยกตัวอย่างเปรียบเทียบความแตกต่างระหว่าง UX / UI กันแบบง่าย ๆ ก็ให้ท่านลองนึกถึงการออกแบบรองเท้าสักหนึ่งคู่ การที่เราคำนึงถึงการออกแบบ ลวดลาย การดีไซน์ออกมาให้ดูสวยเก๋ รวมถึงการใช้วัสดุที่ดีมีคุณภาพ (ตรงนี้คือ UI) และการที่เราคำนึงถึงความพึงพอใจของลูกค้า อยากให้ลูกค้าใส่แล้วรู้สึกเบา สบาย สวมใส่ง่าย รับผิดชอบต่อสรีระ และไม่กัดเท้า (ตรงนี้คือ UX) นี่คือการพิจารณาของการออกแบบทุกสรรพสิ่งในโลก (UX / UI คืออะไร?, 2019)

บทที่ 3

วิธีดำเนินการวิจัย

ปริญญานิพนธ์นี้มีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพการควบคุมการผลิตโดยแสดงสถานะการผลิตต่อวัน รวมทั้งแสดงเวลามาตรฐานและแสดงสถานะการทำงานใน โดยจากการศึกษาสภาพปัญหาภายในโรงงานนั้นผู้วิจัยได้พบว่าสายการผลิต ไม่มีหน้าจอแสดงผลสถานะการผลิตต่อวันตามมาตรฐานโรงงานโดยทั่วไป ดังนั้นผู้วิจัยจึงมีความต้องการที่จะแก้ปัญหานี้โดยการออกแบบหน้าจอแสดงผลและการสร้างโปรแกรมสำหรับควบคุมสถานะการผลิตโดยสามารถแบ่งเป็นขั้นตอนได้ดังต่อไปนี้

3.1 ประชากรที่ใช้ในการศึกษา

ประชากรที่ใช้ในการศึกษาวิจัยนี้เป็นการเลือกแบบเจาะจง คือพนักงานในสายการประกอบ RTD สถานีทำงาน Main line 10 บริษัท อาร์เอ็มเอ กรุ๊ป จำกัด จำนวน 5 คน

3.1.2 เครื่องมือที่ใช้ในงานวิจัย

ผู้วิจัยได้นำความรู้เกี่ยวกับภาษาคอมพิวเตอร์มาใช้พัฒนา ร่วมกับโปรแกรมที่เกี่ยวข้อง เพื่อช่วยการจัดเก็บข้อมูล การค้นหาข้อมูล และวิเคราะห์ปัญหาจากข้อมูล เครื่องมือที่นำมาใช้ในงานวิจัยมีดังนี้

3.1.2.1 วัสดุอุปกรณ์

1. คอมพิวเตอร์ในการพัฒนาโปรแกรม
2. สมาร์ททีวี (Smart TV)
3. เครื่องอ่านรหัสคิวอาร์ (QR-Code Scanner)

3.1.2.2 ซอฟต์แวร์

1. โปรแกรมพีเอชพีมายแอตมิน (phpMyAdmin) สำหรับจัดการฐานข้อมูล
2. โปรแกรมแอปเซิร์ฟ (AppServ)
3. โปรแกรมวิซวลสตูดิโอ ดอทเน็ต (Visual Studio .NET)
4. โปรแกรมเอ็กเซล (Excel)

3.2 การศึกษาและรวบรวมข้อมูล

3.2.1 การศึกษาและรวบรวมข้อมูลสายการประกอบอาที่ดี

จากการสำรวจข้อมูลสายการประกอบรถยนต์ พบว่าในปัจจุบันสายการผลิตของบริษัท RMA นั้นยังคงใช้การแสดงสถานะการผลิตโดยการแขวนป้ายบอกหมายเลข Lot และจำนวนยอดการผลิตในเวลาปัจจุบัน ดังภาพที่ 3.1-3.3

Project Nissan		Station		Date: 14/01/2020	
Target (เป้าหมาย)		Production Output (ผลการผลิตต่อวัน)			
No. OT 12 UPD	1	1:00	1:00	1:00	1:00
	2	1:00	1:00	1:00	1:00
	3	1:00	1:00	1:00	1:00
	4	1:00	1:00	1:00	1:00
	5	1:00	1:00	1:00	1:00
	6	1:00	1:00	1:00	1:00
	7	1:00	1:00	1:00	1:00
	8	1:00	1:00	1:00	1:00
	9	1:00	1:00	1:00	1:00
	10	1:00	1:00	1:00	1:00
	11	1:00	1:00	1:00	1:00
	12	1:00	1:00	1:00	1:00
OT 15 UPD	13	1:00	1:00	1:00	1:00
	14	1:00	1:00	1:00	1:00
	15	1:00	1:00	1:00	1:00

5 คัน/วัน

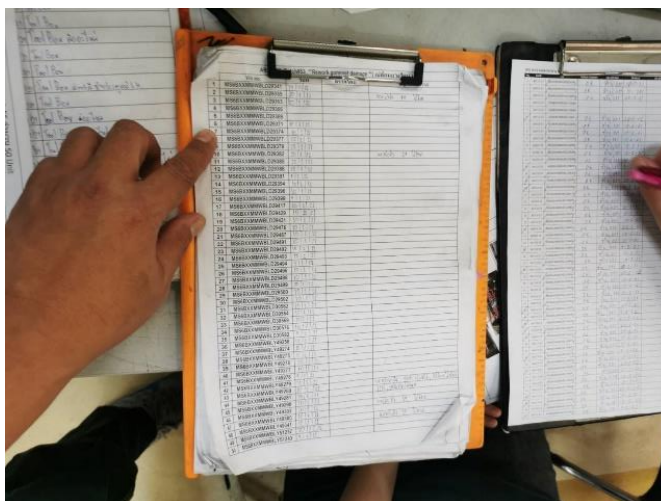
* รหัสบาร์ "รีด" 17:40:16

คัน/วัน

ภาพที่ 3.1 กระดานบอกสถานะการผลิตต่อวัน



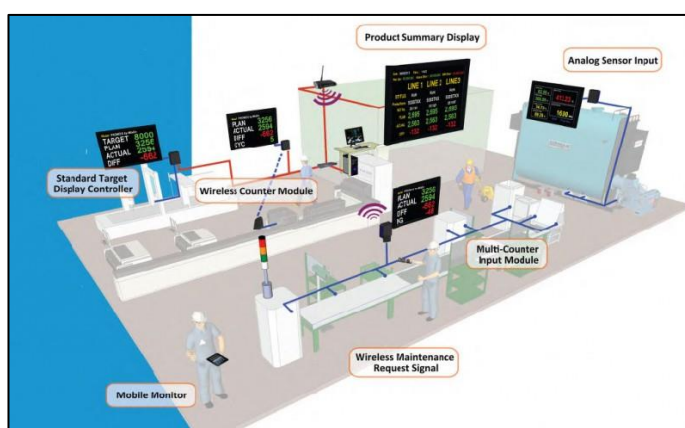
ภาพที่ 3.2 ป้ายแขวนบอกตัวเลข Lot และยอดการผลิตในเวลาปัจจุบัน



ภาพที่ 3.3 เช็คยอดการผลิตตามหมายเลขประจำรถยนต์

3.3 เสนอแนวทางการติดตั้ง

จากการกำหนดปัญหาที่จะแก้ไขและตั้งเป้าหมาย ทางคณะผู้วิจัยจึงได้นำเสนอการติดตั้งจอแสดงสถานะควบคุมการผลิตเพื่อแสดงยอดเป้าหมายการผลิตต่อวัน การกำหนดเวลามาตรฐาน และการแสดงผลเมื่อเกิดการหยุดการทำงานของสายการผลิตพร้อมทั้งบันทึกข้อมูลรายละเอียดการผลิตยนต์ตามหมายเลขประจำรถยนต์ (Vin Number) ดังภาพที่ 3.4



ภาพที่ 3.4 ระบบควบคุมการแสดงผลภายในสายการผลิต

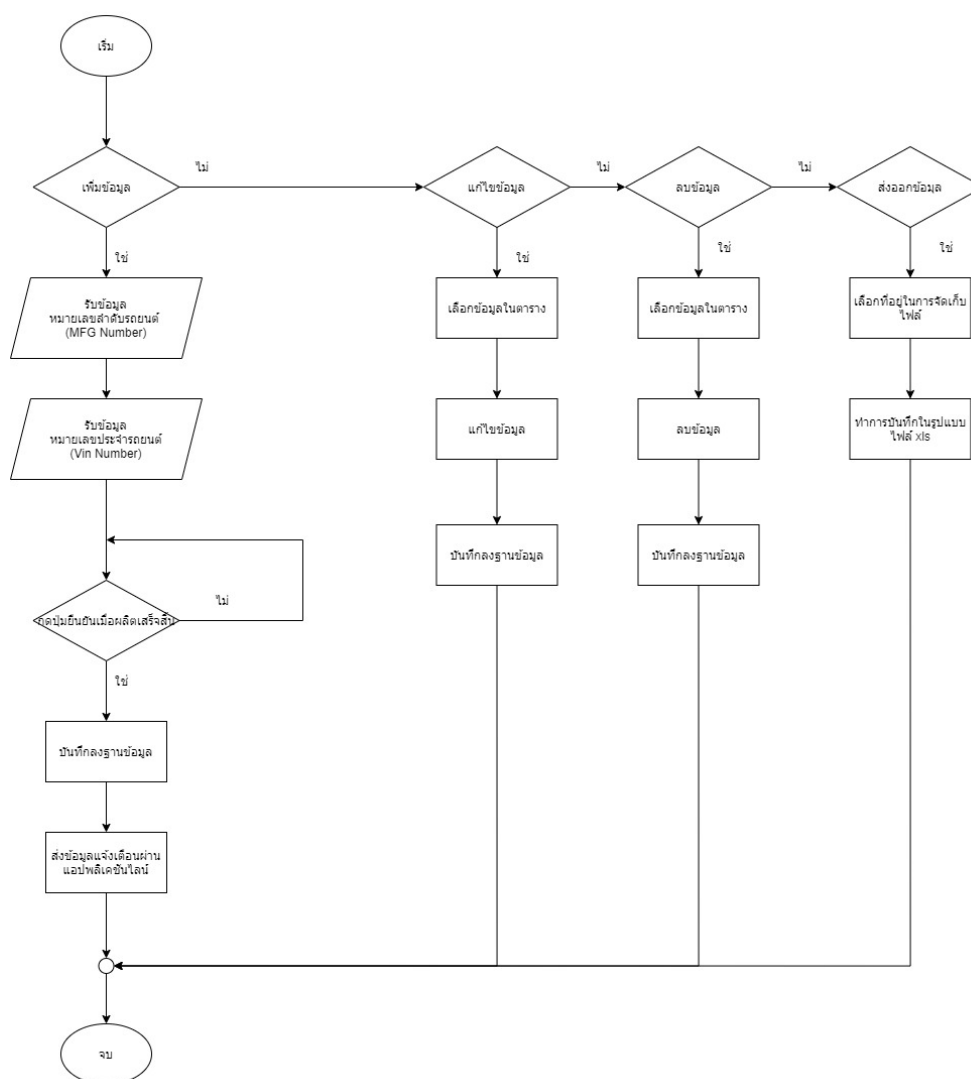
3.4 วิเคราะห์และออกแบบระบบ

3.4.1. การออกแบบแผนภาพแสดงการทำงานของโปรแกรมแสดงผลและควบคุมสถานการณ์การผลิต

แผนภาพแสดงการทำงานของโปรแกรมแสดงผลและควบคุมสถานการณ์การผลิตแสดงให้เห็นถึงภาพรวมการทำงานของแต่ละหน้า โดยมีแผนภาพแสดงขั้นตอนการทำงานดังนี้

3.4.1.1 แผนภาพแสดงขั้นตอนการทำงานหน้า MFG Input

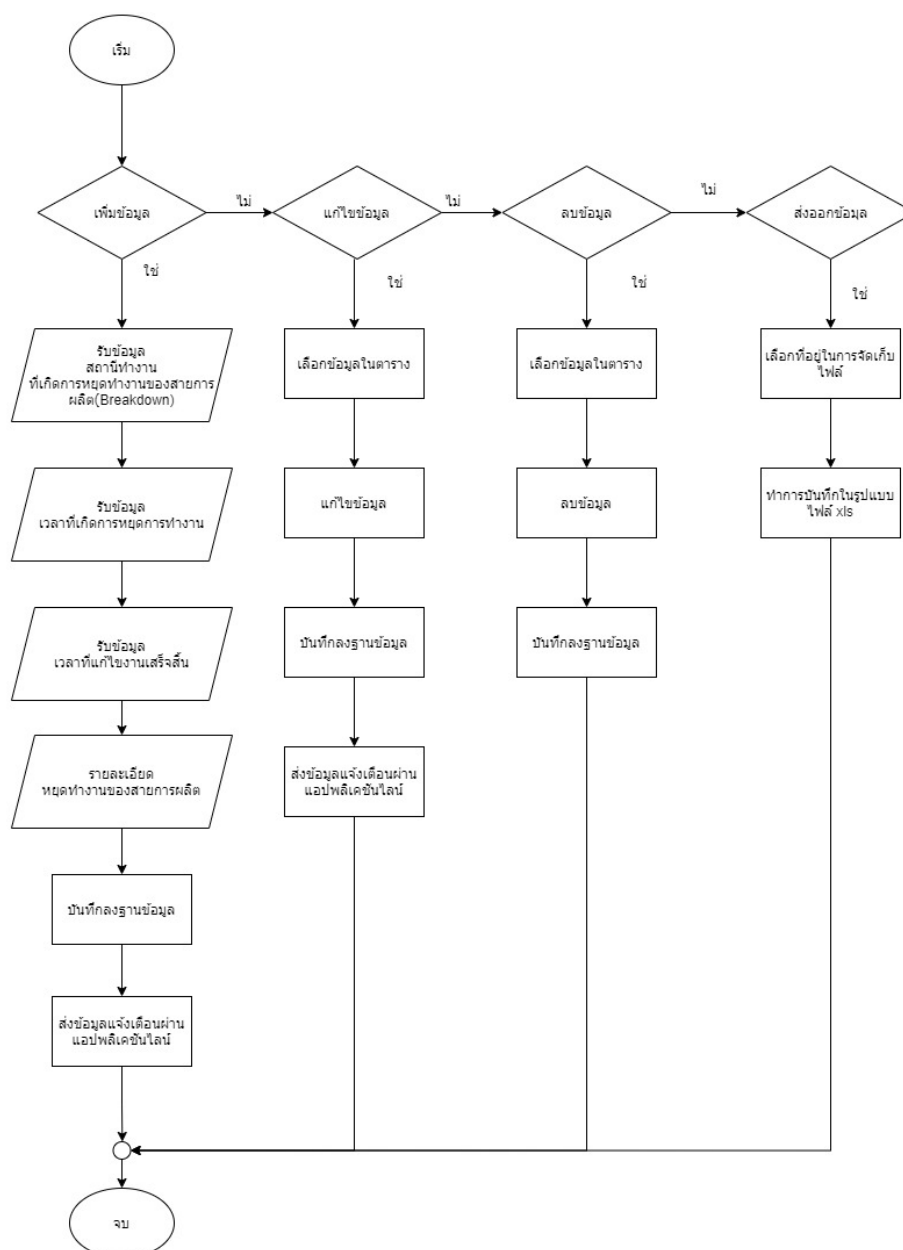
หน้า MFG Input มีหน้าที่ใช้ในการ เพิ่ม แก้ไข ลบ และทำการคำนวณยอดการผลิตในแต่ละวัน จากนั้นโปรแกรมจะทำการบันทึกลงในฐานข้อมูล ดังภาพ 3.5



ภาพที่ 3.5 แผนภาพแสดงขั้นตอนการทำงานหน้า MFG Input

3.4.1.2 แผนภาพแสดงขั้นตอนการทำงานหน้า Breakdown Input

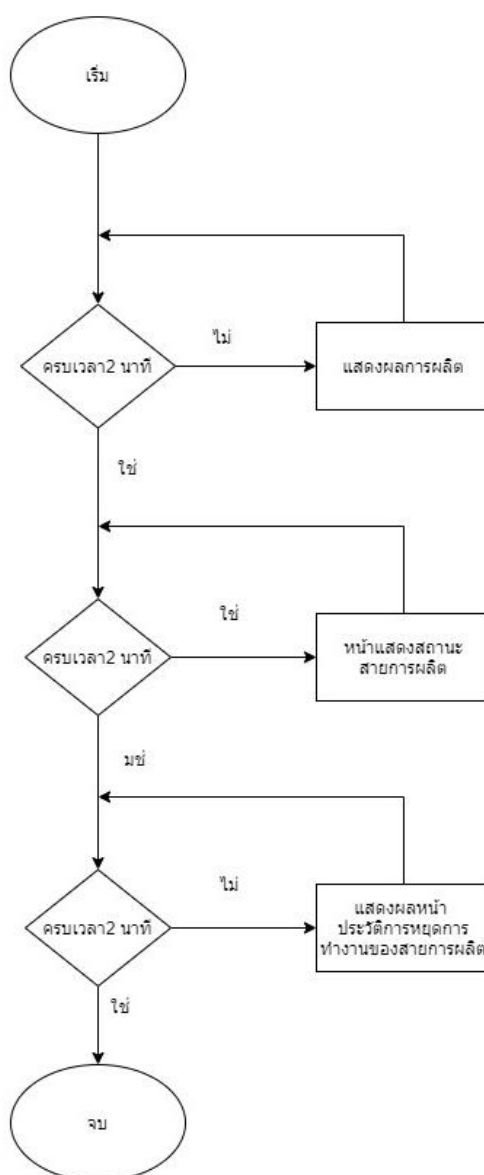
หน้า Breakdown Input มีหน้าที่ใช้ในการ เพิ่ม แก้ไข ลบ การเกิดการหยุดการทำงานของสายการผลิตในแต่ละวันจากนั้นโปรแกรมจะทำการบันทึกลงในฐานข้อมูล ดังภาพ 3.6



ภาพที่ 3.6 แผนภาพแสดงขั้นตอนการทำงานหน้า Breakdown Input

3.4.1.3 แผนภาพแสดงลำดับหน้าจอที่แสดงผลข้อมูลในสายการผลิต

ลำดับหน้าจอแสดงผลในสายการผลิต โดยแบ่งออกเป็น 3 หน้าด้วยกัน คือ 1. หน้าแสดงผลการผลิต 2. หน้าแสดงสถานะสายการผลิต 3. หน้าประวัติที่เกิดการหยุดการทำงาน ของสายการผลิต ดังภาพที่ 3.7



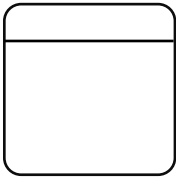



ภาพที่ 3.7 แผนภาพแสดงลำดับหน้าจอแสดงผลในสายการผลิต

3.4.2 การออกแบบระบบฐานข้อมูล

เมื่อรวบรวมข้อมูลเรียบร้อยแล้วก็นำข้อมูลมาทำให้เป็นระบบและวิเคราะห์ หาความเป็นไปได้ในการจัดทำระบบ โดยออกแบบระบบดังนี้

การออกแบบแผนภาพการไหลของข้อมูล(Data Flow Diagram) เป็นการอธิบายถึงการไหลเวียนของข้อมูลและกระบวนการทำงานที่เกี่ยวข้องในการจัดการพัฒนาข้อมูลระบบฐานข้อมูล โดยมีสัญลักษณ์ในการอธิบายการไหลของข้อมูล 4 สัญลักษณ์ ดังตารางที่ 3.1

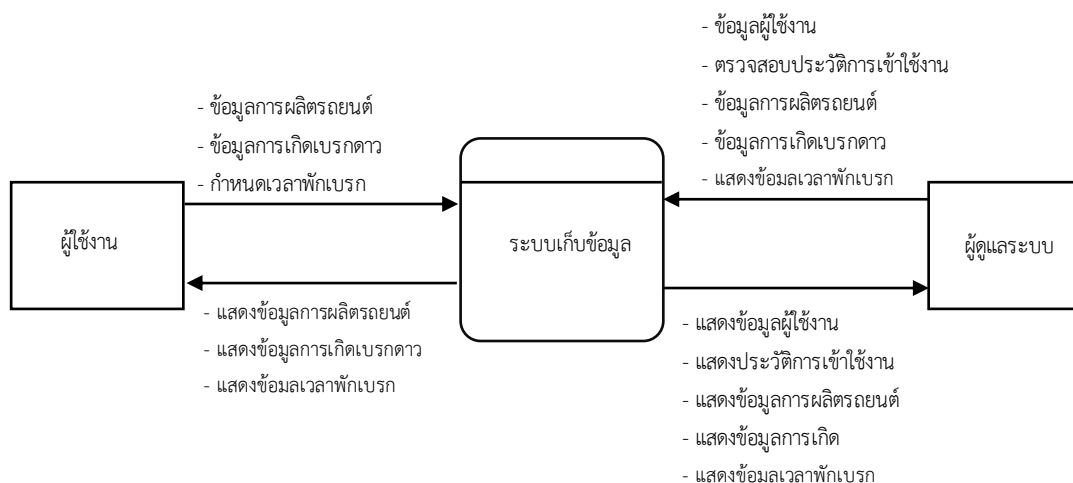
ตารางที่ 3.1 แสดงแผนภาพการไหลของข้อมูล

สัญลักษณ์	ชื่อ	คำอธิบาย
	Process symbol	แทนด้วยภาพสี่เหลี่ยมมุมมน มีหน้าที่รับข้อมูลและจัดการกับข้อมูลแล้วส่งผลลัพธ์ของข้อมูลดังกล่าวที่ผ่านการประมวลผล
	Data Flow symbol	แสดงด้วยเส้นลูกศรและกำกับด้วยชื่อของข้อมูลจะเป็นการเชื่อมโยงระหว่าง Process symbol กับสิ่งอื่น
	Data store symbol	แสดงด้วยสี่เหลี่ยมที่เปิดหนึ่งด้าน เป็นการแสดงการเก็บข้อมูลที่จะนำมาประมวลผลภายหลังต้องมีการเชื่อมต่อ Process โดย Data Flow
	External Entity Symbol	(หน่วยภายนอก) แสดงด้วยสี่เหลี่ยม เป็นการแสดงถึงคน หน่วยงาน องค์กร หรือบริษัท ที่เกี่ยวข้องกับระบบ

3.4.3 การพัฒนาระบบฐานข้อมูลโปรแกรมแสดงผลและควบคุมสถานการณ์ผลิต

การสร้างระบบฐานข้อมูลโปรแกรมแสดงผลและควบคุมสถานการณ์ผลิต มีรายละเอียดขั้นตอนการสร้างดังต่อไปนี้

3.4.3.1 แผนภาพบริบท Context Diagram



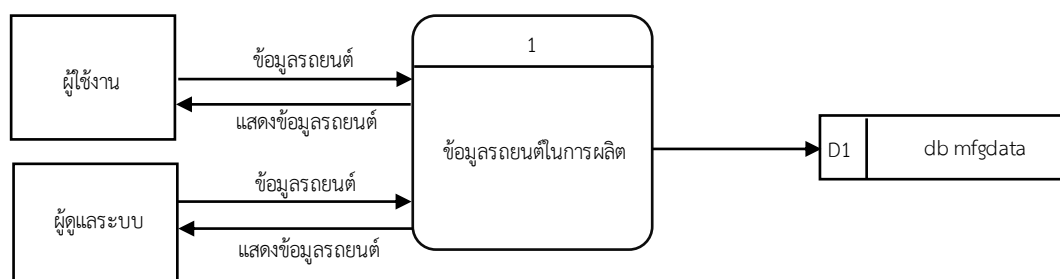
ภาพที่ 3.8 แผนภาพบริบท Context Diagram

จากภาพที่ 3.8 รายละเอียด แผนภาพบริบท (Context Diagram) ระบบแสดงผลข้อมูลและควบคุมการผลิตมีเอนทิตีหลัก 2 เอนทิตีดังนี้

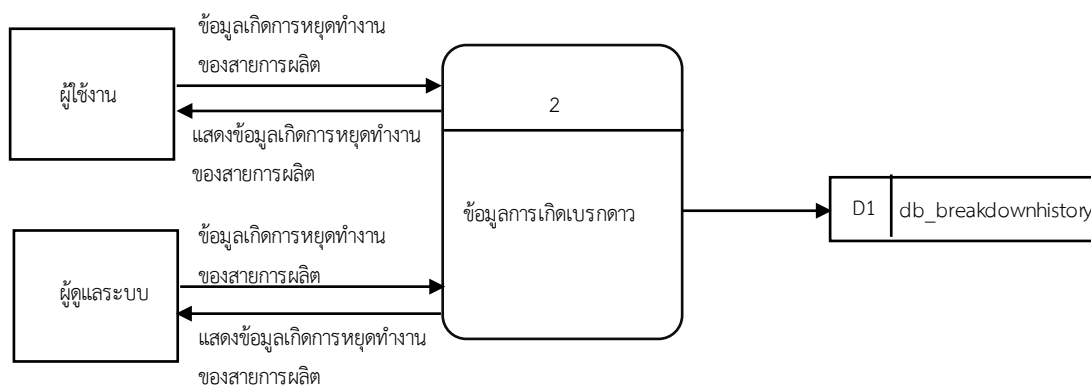
พนักงาน พนักงานสามารถ เพิ่ม ลบ แก้ไขข้อมูลการผลิตรถยนต์ต่อวันและสามารถแก้ไขข้อมูลเมื่อเกิดการหยุดการทำงานของสายการผลิต

ผู้ดูแลระบบ ผู้ดูแลสามารถ เพิ่ม ลบ แก้ไข ไม่ว่าจะเป็น ข้อมูลผู้ใช้งาน ข้อมูลการผลิตรถยนต์และสามารถแก้ไขข้อมูลเมื่อเกิดการหยุดการทำงานของสายการผลิต

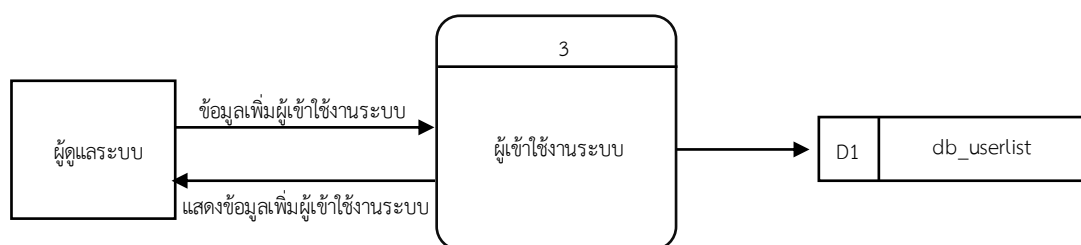
3.4.3.2 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1)



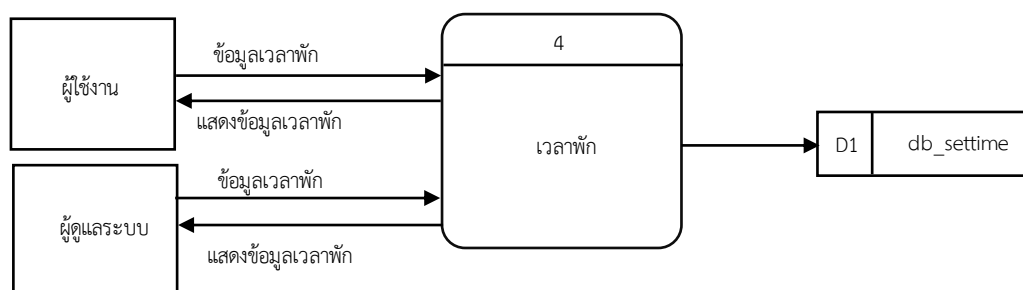
ภาพที่ 3.9 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1) ในส่วนของข้อมูลรถยนต์ในการผลิต



ภาพที่ 3.10 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1) ในส่วนของข้อมูล
ข้อมูลการหยุดการทำงานของสายการผลิต

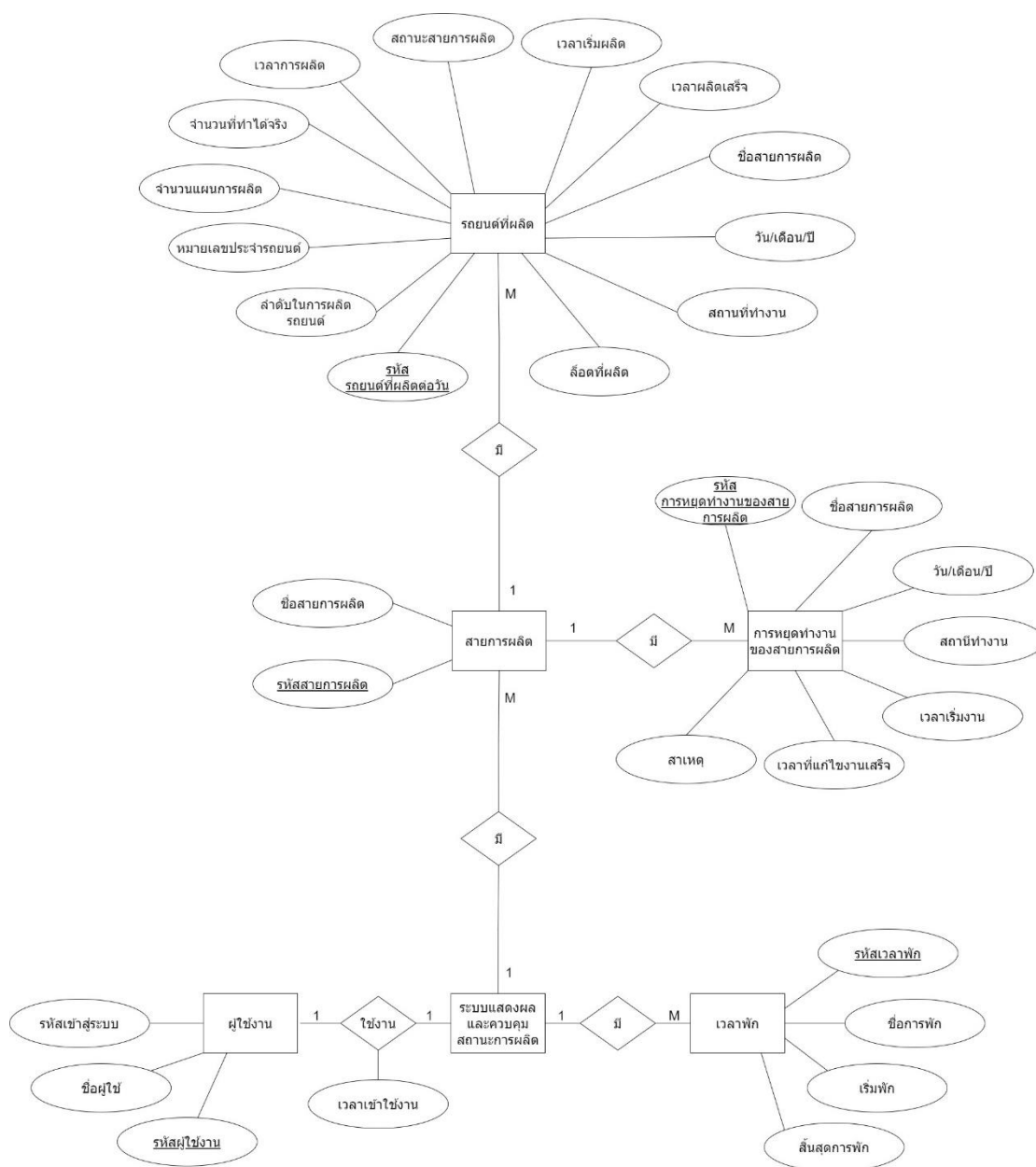


ภาพที่ 3.11 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1) ในส่วนของผู้เข้า
ใช้งานระบบ



ภาพที่ 3.12 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1) ในส่วนของ
เวลาพัก

3.4.3.3 แผนภาพแสดงความสัมพันธ์ของข้อมูล (Entity - Relationship Diagram)



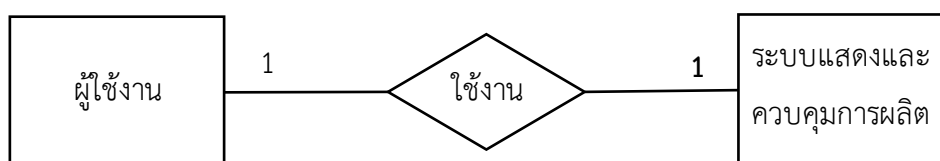
ภาพที่ 3.13 แผนภาพแสดงความสัมพันธ์ของข้อมูล (Entity Relationship Diagram) ในส่วนของระบบจัดเก็บข้อมูลการแสดงผลและควบคุมสถานะการผลิต

3.4.4 ความสัมพันธ์ของตาราง

การออกแบบสำหรับพัฒนาระบบ ต้องอาศัยเทคนิคการออกแบบฐานข้อมูลเชิงสัมพันธ์ โดยใช้โมเดล E-R(Entity Relation Model) เพื่อแสดงความสัมพันธ์ระหว่างเอนทิตี ซึ่งมีรายละเอียดดังนี้

3.4.4.1 ความสัมพันธ์ระหว่าง ตารางประเภทผู้ใช้งาน กับ ตารางระบบแสดงและควบคุมการผลิต

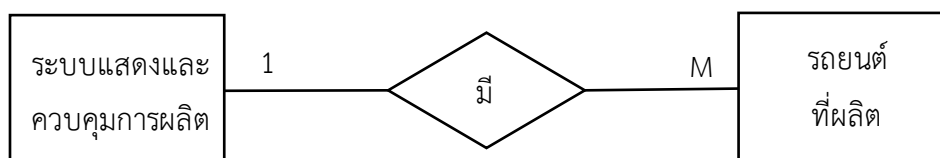
จากภาพที่ 3.14 ผู้ใช้งาน 1 คนสามารถใช้งานระบบแสดงและควบคุมการผลิตได้ 1 ระบบ



ภาพที่ 3.14 แสดงความสัมพันธ์ระหว่าง ตารางประเภทผู้ใช้งาน กับ ตารางระบบแสดงและควบคุมการผลิต

3.4.4.2 ความสัมพันธ์ระหว่าง ตารางประเภทระบบแสดงและควบคุมการผลิต กับ ตารางรถยนต์ที่ผลิตต่อวัน

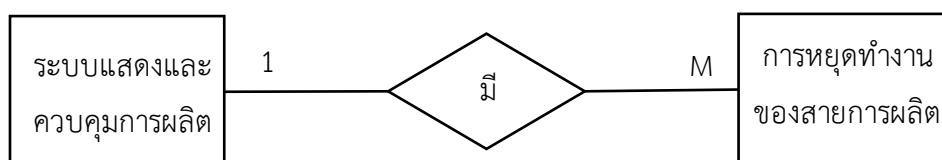
จากภาพที่ 3.15 ระบบแสดงและควบคุมการผลิต 1 ระบบ สามารถมี รถยนต์ที่ผลิตได้หลายคัน รถยนต์ที่ผลิต 1 คัน สามารถมีระบบแสดงและควบคุมการผลิตได้ 1 ระบบ



ภาพที่ 3.15 แสดงความสัมพันธ์ระหว่าง ตารางประเภทระบบแสดงและควบคุมการผลิต กับ ตารางรถยนต์ที่ผลิตต่อวัน

3.4.4.3 ความสัมพันธ์ระหว่าง ตารางประเภทระบบแสดงและควบคุมการผลิต กับ ตารางการหยุดทำงานของสายการผลิต

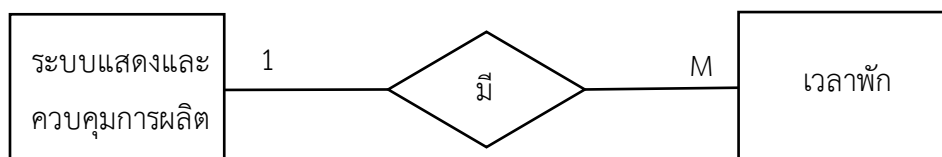
จากภาพที่ 3.16 ระบบแสดงและควบคุมการผลิต 1 ระบบ สามารถมีการหยุดทำงานของสายการผลิตได้หลายครั้ง การหยุดทำงานของสายการผลิต 1 ครั้ง สามารถมีระบบแสดงและควบคุมการผลิตได้ 1 ระบบ



ภาพที่ 3.16 ความสัมพันธ์ระหว่าง ตารางประเภทระบบแสดงและควบคุมการผลิต กับ ตารางการหยุดทำงานของสายการผลิต

3.4.4.4 ความสัมพันธ์ระหว่าง ตารางประเภท ระบบแสดงและควบคุมการผลิต กับ ตารางเวลาพัก

จากภาพที่ 3.17 ระบบแสดงและควบคุมการผลิต 1 ระบบ สามารถมีเวลาพักได้หลายครั้ง เวลาพัก 1 ครั้ง สามารถมีระบบแสดงและควบคุมการผลิตได้ 1 ระบบ



ภาพที่ 3.17 แสดงความสัมพันธ์ระหว่าง ตารางประเภทระบบแสดงและควบคุมการผลิต กับ ตารางเวลาพัก

3.4.4.5 ตารางข้อมูล

ตารางที่ 3.2 ข้อมูลการผลิตรถยนต์ต่อวัน

ข้อมูลการผลิตรถยนต์ต่อวัน				
ลำดับ	Attribute	Descriptions	Type	PK
1	Id	รหัสข้อมูลการผลิต	Int(11)	PK
2	State	สายการผลิต	Varchar(20)	
3	Date	วันที่บันทึก	Varchar(20)	
4	Time_Start	เวลาที่เริ่มผลิต/คัน	Varchar(20)	
5	Time_Finish	เวลาที่ผลิตเสร็จ/คัน	Varchar(20)	
6	Lot	จำนวนล็อตที่ผลิต	Varchar(100)	
7	Vin	หมายเลขรถยนต์	Varchar(100)	
8	Mfg	จำนวนรถที่ผลิต	Varchar(100)	
9	Plan	จำนวนที่วางแผนการผลิต	Varchar(100)	
10	Actual	จำนวนที่ทำได้จริง	Varchar(100)	
11	TakeTime	เวลาในการผลิต	Varchar(100)	
12	Status	สถานะสายการผลิต	Varchar(100)	

ตารางที่ 3.3 ข้อมูลการหยุดทำงานของสายการผลิต

ข้อมูลการหยุดทำงานของสายการผลิต				
ลำดับ	Attribute	Descriptions	Type	PK
1	Id	รหัสข้อมูลการผลิต	Int(11)	PK
2	Date	วัน/เดือน/ปี	Varchar(20)	
3	StartBreakdown	เวลาเริ่ม	Varchar(20)	
4	FinishBreakdown	เวลาสิ้นสุด	Varchar(20)	

ตารางที่ 3.4 ข้อมูลการหยุดทำงานของสายการผลิต (ต่อ)

ข้อมูลการหยุดทำงานของสายการผลิต				
ลำดับ	Attribute	Descriptions	Type	PK
5	TotalBreakdown	เวลาทั้งหมดในการแก้ไข	Varchar(20)	
6	Detail	รายละเอียดการเกิดปัญหา	Varchar(100)	
7	Lineproduction	สายการผลิต	Varchar(100)	
8	Station	จำนวนรถที่ผลิต	Varchar(100)	
9	Status	จำนวนที่วางแผนการผลิต	Varchar(100)	

ตารางที่ 3.5 ข้อมูลเวลาการพัก

ข้อมูลเวลาการพัก				
ลำดับ	Attribute	Descriptions	Type	PK
1	Id	รหัสข้อมูลการผลิต	Int(11)	PK
2	Name	ชื่อการพัก	Varchar(20)	
3	Starttime	เวลาเริ่ม	Varchar(20)	
4	Finishtime	เวลาสิ้นสุด	Varchar(20)	

ตารางที่ 3.6 ข้อมูลผู้ใช้งาน

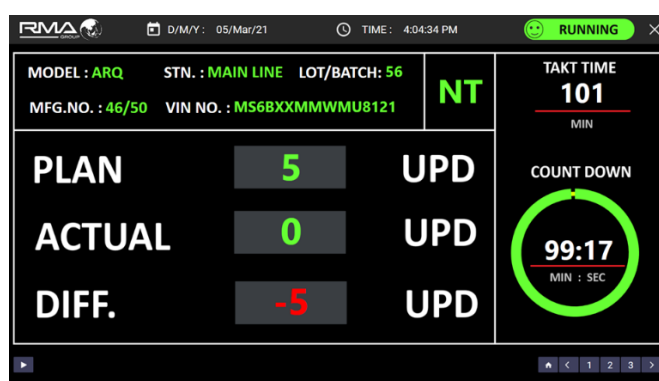
ข้อมูลผู้ใช้งาน				
ลำดับ	Attribute	Descriptions	Type	PK
1	Id	รหัสผู้ใช้งาน	Int(11)	PK
2	Name	ชื่อผู้ใช้งาน	Varchar(100)	
3	Password	รหัสเข้าใช้งาน	Varchar(100)	

3.5 การออกแบบหน้าจอแสดงผลข้อมูล

การออกแบบหน้าจอแสดงผลข้อมูล ผู้วิจัยได้ออกแบบให้มีความเหมาะสมกับผู้ใช้งานคำนึงถึงความสะดวกและความต้องการของผู้ใช้เป็นหลัก โดยแบ่งออกเป็น 3 ส่วนดังนี้

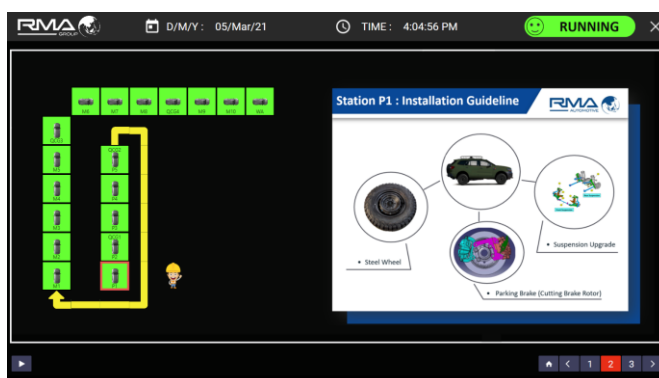
3.5.1 การแสดงผลข้อมูลในสายการผลิต

ในส่วนของการแสดงผลข้อมูลในสายการผลิตได้ถูกแบ่งออกเป็น 4 หน้า ประกอบไปด้วย หน้าแสดงข้อมูลการผลิต หน้าแสดงสถานะสายการผลิต หน้าประวัติการหยุดการทำงานของสายการผลิต หน้าแสดงเวลาพักของพนักงาน



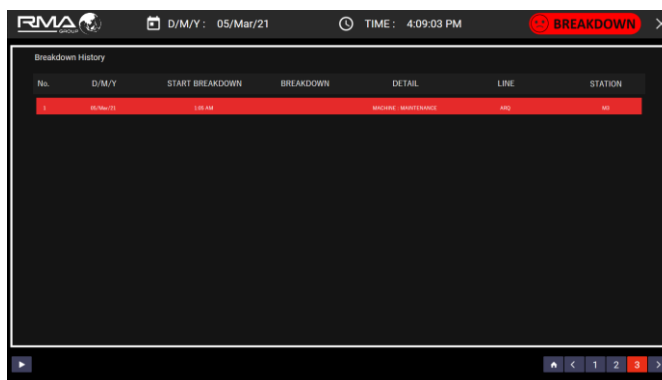
ภาพที่ 3.18 ตัวอย่างหน้าจอแสดงผลการผลิต

จากภาพที่ 3.18 แสดงหน้าจอแสดงผลการผลิต (MFG Output) โดยจะแสดง ข้อมูลต่างๆในการผลิต เช่น ชื่อโมเดลการผลิต (Model Name) จำนวนรถยนต์ที่ผลิต (MFG.No), จำนวนที่วางแผนการผลิต (Plan),จำนวนที่ทำได้จริง (Actual),เวลาที่ใช้ในการผลิต เป็นต้น



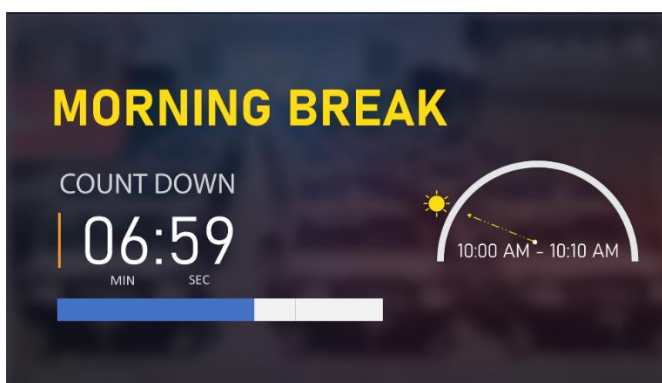
ภาพที่ 3.19 ตัวอย่างหน้าจอแสดงสถานะสายการผลิต

จากภาพที่ 3.19 แสดงภาพตัวอย่างในส่วนของหน้าแสดงสายการผลิต โดยจะแสดง สถานีทำงานในสายการผลิต RTD



ภาพที่ 3.20 ภาพการออกแบบหน้าแสดงผลประวัติการหยุดการทำงานของสายการผลิต

จากภาพที่ 3.20 แสดงตัวอย่างหน้าแสดงผลประวัติการหยุดการทำงานของสายการผลิต ต่อวันโดยประกอบไปด้วยข้อมูล วัน เดือน ปี, เวลาที่เริ่มหยุดการทำงานของสายการผลิต, เวลาที่เกิดการหยุดการทำงานของสายการผลิต, รายละเอียดในการเกิดการหยุดการทำงานของสายการผลิต



ภาพที่ 3.21 ตัวอย่างแสดงหน้าพักเบรก

จากภาพที่ 3.21 แสดงตัวอย่างหน้าพักเบรกของพนักงานประกอบไปด้วยข้อมูล สถานะการพักเบรกเวลาในการพักเบรก

3.5.2 การควบคุม

ในส่วนของหน้าควบคุมโดยหน้าที่ของหน้านี้คือ เพิ่ม แก้ไข ลบ ข้อมูลต่างๆ ลงในฐานข้อมูลและส่งข้อมูลขึ้นหน้าจอแสดงผลในสายการผลิตโดยแบ่งออกเป็น 3 หน้า ประกอบไปด้วย หน้าป้อนข้อมูลการผลิต (MFG Input) ,หน้าป้อนข้อมูลการหยุดการทำงานของสายการผลิต (Breakdown Input) ,หน้ากำหนดเวลาการพักเบรก (Set Breaktime) ตามภาพตัวอย่างดังนี้

ภาพที่ 3.22 ตัวอย่างหน้าป้อนข้อมูลการผลิต (MFG Input)

ภาพที่ 3.23 ตัวอย่างหน้าป้อนข้อมูลการหยุดการทำงานของสายการผลิต (Breakdown Input)

	Start	End
MorningBreak :	1:53 AM	MorningBreak : 1:55 AM
LunchBreak :	8:25 PM	LunchBreak : 8:30 PM
AfternoonBreak :	7:50 PM	AfternoonBreak : 7:51 PM
OvertimeBreak :	1:20 AM	OvertimeBreak : 1:21 AM

save cancel

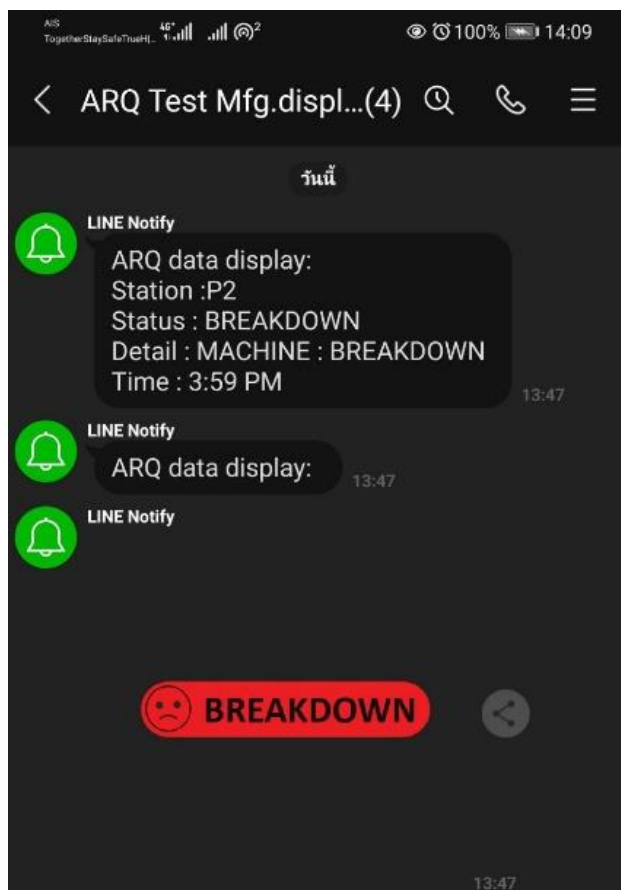
ภาพที่ 3.24 ตัวอย่างหน้ากำหนดเวลาการพักเบรก (Set Breaktime)

3.5.3 การแสดงข้อมูลแจ้งเตือนผ่านแอปพลิเคชันไลน์

ในส่วนของการแจ้งเตือนผ่านแอปพลิเคชันไลน์ ระบบจะส่งข้อมูลเกี่ยวกับการผลิตรถยนต์ต่อวันเข้าไลน์ ประกอบไปด้วย 2 ส่วน ได้แก่ การผลิตรถยนต์ต่อวัน และการหยุดทำงานของสายการผลิตซึ่งในแต่ละส่วนจะแสดงข้อมูล ดังนี้ ชื่อสายการผลิต วันที่ผลิต เวลาที่ผลิตเสร็จ และเลขประจำรถยนต์ (Vin Number)



ภาพที่ 3.25 ตัวอย่างการแจ้งเตือนการผลิตรถยนต์ต่อวันผ่านแอปพลิเคชันไลน์



ภาพที่ 3.26 ตัวอย่างหน้าป้อนข้อมูลการผลิต (MFG Input)

บทที่ 4

ผลการวิจัย

ในการดำเนินการปริญญานิพนธ์เรื่องโปรแกรมแสดงผลและควบคุมสถานะการผลิต คณะผู้วิจัยได้ออกแบบและสร้างโปรแกรมแสดงผลและควบคุมสถานะการผลิต เพื่อเพิ่มประสิทธิภาพการควบคุมการผลิตโดยแสดงสถานะการผลิตต่อวัน รวมถึงการเก็บฐานข้อมูล (Database) เมื่อเกิดการหยุดการทำงานของสายการผลิต (Breakdown) เพื่อให้ง่ายต่อการนำข้อมูลมาวิเคราะห์ โดยผลการวิจัยมีดังนี้

4.1 ผลการดำเนินงานวิจัย

4.1.1 ผลของการสำรวจของประชากร 5 คน ที่เข้าใช้โปรแกรมแสดงผลและควบคุมสถานะการผลิต

ผู้วิจัยได้สำรวจความพึงพอใจประชากร 5 คน ในสายการประกอบ RTD สถานีทำงาน Main line 10 ที่เข้าโปรแกรมแสดงผลและควบคุมสถานะการผลิต โดยมีหัวข้อสำรวจ 4 หัวข้อ ได้แก่ การใช้งานโปรแกรมแสดงผลและควบคุมสถานะการผลิต ความเข้าใจในการดูผลลัพธ์ เนื้อหาที่ครบถ้วน ตัวอักษรที่มองเห็น โดยมีผลดังตารางที่ 4.1

ตารางที่ 4.1 แสดงคะแนนประชากร 5 คน ที่ใช้งานโปรแกรมแสดงผลและควบคุมสถานะการผลิต

จำนวน (คน)	คะแนน เต็ม	ใช้งานง่าย	ความเข้าใจ ในการดู ผลลัพธ์	เนื้อหา ครบถ้วน	สีและขนาด ตัวอักษร
1	5	3	5	5	5
2	5	3	5	5	5
3	5	5	4	5	5
4	5	5	5	5	5
5	5	4	4	5	5

จากตารางที่ 4.1 แสดงผลสำรวจความพึงพอใจจากประชากร 5 คน ในสายการประกอบ RTD สถานีทำงาน Main line 10 พบว่าคะแนนในหัวข้อใช้งานง่าย มีประชากรจำนวน 2 คน ให้คะแนนอยู่ใน

ระดับ 3 ประชากรจำนวน 1 คน ให้คะแนนอยู่ในระดับ 4 และมีประชากรจำนวน 2 คน ให้คะแนนอยู่ในระดับ 5 เฉลี่ยคะแนนในหัวข้อใช้งานง่ายเท่ากับ 4 คะแนน

คะแนนในหัวข้อความเข้าใจในการดูผลลัพธ์ มีประชากรจำนวน 3 คน ให้คะแนนอยู่ในระดับ 5 ประชากรจำนวน 2 คน ให้คะแนนอยู่ในระดับ 4 เฉลี่ยคะแนนในหัวข้อความเข้าใจในการดูผลลัพธ์ใช้งานง่ายเท่ากับ 4.6 คะแนน

คะแนนในหัวข้อเนื้อหาครบถ้วน มีประชากรจำนวน 5 คน ให้คะแนนอยู่ในระดับ 5 เฉลี่ยคะแนนในหัวข้อเนื้อหาครบถ้วนเท่ากับ 5 คะแนน

คะแนนในหัวข้อสีและขนาดตัวอักษร มีประชากรจำนวน 5 คน ให้คะแนนอยู่ในระดับ 5 เฉลี่ยคะแนนในหัวข้อสีและขนาดตัวอักษร เท่ากับ 5 คะแนน

4.1.2 การเปรียบเทียบการทำงานแบบเดิมและแบบใหม่

ผลจากการพัฒนาโปรแกรมแสดงผลและควบคุมสถานะการผลิตนี้ ได้ทำการสรุปและเปรียบเทียบการทำงานแบบเดิมและแบบใหม่ โดยมีตัวชี้วัดจากหัวข้อได้แก่ การบันทึกผลการทำงาน การค้นหาข้อมูลที่ต้องการและการแสดงข้อมูลขณะทำงาน ดังตารางที่ 4.2

ตารางที่ 4.2 แสดงการเปรียบเทียบการทำงานแบบเดิมและแบบใหม่

การทำงาน	การบันทึกข้อมูล	การค้นหาข้อมูล	การแสดงผลข้อมูล	การแจ้งเตือน
แบบเดิม	กรอกใส่กระดาษ	ค้นหาจากกระดาษที่บันทึก	กระดานและป้าย	ไม่มีการแจ้งเตือน
แบบใหม่	คีย์เข้าโปรแกรม	ค้นหาผ่านโปรแกรม	แดชบอร์ด	ผ่านแอปพลิเคชันไลน์

จากตารางที่ 4.2 สรุปได้ว่าการทำงานแบบเดิม พนักงานจะต้องกรอกข้อมูลรถยนต์ที่ผลิตต่อวันและปัญหาการหยุดทำงานของสายการผลิตใส่กระดาษในส่วนของการค้นหาข้อมูลพนักงานจะต้องค้นหาจากกระดาษที่บันทึกไว้ซึ่งยากต่อการค้นหาและติดตามข้อมูลและสุดท้ายสำหรับการแสดงผลจะแสดงผลบนกระดานและป้ายเพื่อแจ้งให้พนักงานในสายการผลิตทราบได้ถึงข้อมูล

ซึ่งการทำงานแบบใหม่พนักงานจะสามารถกรอกข้อมูลรถยนต์ที่ผลิตต่อวันและปัญหาการหยุดทำงานของสายการผลิตใส่ในโปรแกรมได้ทันทีที่สามารถค้นหาข้อมูลโดยง่ายพร้อมทั้งแจ้งเตือนผ่านแอปพลิเคชันไลน์

ตารางที่ 4.3 แสดงระยะเวลาเปรียบเทียบการทำงานแบบเดิมและแบบใหม่

	การทำงานแบบเดิม ระยะเวลาเฉลี่ย (นาทิต่อวัน)	การทำงานแบบใหม่ ระยะเวลาเฉลี่ย (นาทิต่อวัน)	ผลต่าง ของเวลาการทำงาน	
			(นาทิต่อวัน)	เปอร์เซ็นต์
การบันทึกข้อมูล	480	460	20	4.17%
การค้นหาข้อมูล	30	2	28	93.33%
เวลาในการรับรู้ ข้อมูล	60	10	50	83.33%
รวม	570	472	98	17.19%

จากตารางที่ 4.3 สรุปได้ว่าการทำงานแบบใหม่ในการบันทึกข้อมูลสามารถลดเวลาในการทำงานลงได้ 20 นาทิต่อวันหรือคิดเป็น 4.17 เปอร์เซ็นต์ ในการค้นหาข้อมูลสามารถลดเวลาลงได้ 28 นาทิต่อวันหรือคิดเป็น 93.33 เปอร์เซ็นต์ สำหรับเวลาในการรับรู้ข้อมูลสามารถลดระยะเวลาลงได้ 50 นาทิต่อวันหรือคิดเป็น 83.33 เปอร์เซ็นต์ ดังนั้นจากระยะเวลารวมทั้งหมดสามารถลดเวลาลงได้ 98 นาที หรือ คิดเป็น 17.19 เปอร์เซ็นต์ต่อวัน

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

ผลจากการดำเนินงานวิจัยเรื่อง โปรแกรมแสดงผลและควบคุมสถานการณ์ผลิตสามารถช่วยลดระยะเวลาลงได้ 98 นาที หรือ คิดเป็น 17.19 เปอร์เซ็นต์ต่อวันและเพิ่มประสิทธิภาพการควบคุมการผลิตโดยแสดงสถานะการผลิตต่อวันรวมทั้งสามารถแสดงเวลามาตรฐาน แสดงสถานะการทำงานในสายการผลิต สามารถเก็บข้อมูลการผลิตต่อวันและการเกิดเบรกดาวเพื่อให้ง่ายต่อการนำข้อมูลมาวิเคราะห์เมื่อเกิดปัญหาขึ้นภายในสายการผลิต

5.2 ประโยชน์ของวิจัย

1. เป็นการลดปัญหาและข้อผิดพลาดที่เปิดจากการจัดเก็บข้อมูลโดยมนุษย์
2. เพิ่มประสิทธิภาพในการสืบค้นข้อมูลของหน่วยงานใด
3. มีการแสดงผลข้อมูลที่เข้าใจง่ายและตรงตามความต้องการของผู้ใช้งาน
4. อำนวยความสะดวกให้พนักงานการเรียกดูข้อมูลที่ต้องการ
5. ประหยัดงบประมาณเนื่องจากการเก็บข้อมูลลงฐานข้อมูลแทนการใช้กระดาษ

5.3 ปัญหาที่พบ

1. ปัญหาด้านความเหมาะสมในการออกแบบหน้าจอแสดงผลเพื่อให้ง่ายต่อการมองเห็น
2. การแสดงข้อมูลให้ผู้ใช้งานเข้าใจง่าย
3. อุปกรณ์เสริมในการแสดงผล

5.4 แนวทางในการแก้ปัญหา

1. ศึกษาการออกแบบโปรแกรมแสดงผลและควบคุมสถานการณ์ผลิต
2. ศึกษาภาษาที่นำมาพัฒนาโปรแกรมแสดงผลและควบคุมสถานการณ์ผลิต
3. ศึกษาขั้นตอนของการเพิ่มข้อมูลการผลิตรถยนต์

5.5 ข้อเสนอแนะ

1. โปรแกรมสามารถพัฒนาการบันทึกข้อมูลที่ปัจจุบันบันทึกข้อมูลแบบ Local เปลี่ยนเป็นบันทึกข้อมูลแบบ Online โดยใช้ server ของทางบริษัทได้
2. โปรแกรมสามารถวาดสายการผลิตเองได้เพื่อยืดหยุ่นต่อการใช้งาน
3. โปรแกรมสามารถส่งข้อมูลไปยัง Email ได้
4. ติดไฟสถานะการแจ้งเตือนในสายการผลิต

บรรณานุกรม

- Admin ITGenius. (19 กันยายน 2014). *(MySQL) คืออะไร*. เข้าถึงได้จาก itGENIUS:
[https://www.itgenius.co.th/article/\(MySQL\)%20คืออะไร.html](https://www.itgenius.co.th/article/(MySQL)%20คืออะไร.html)
- Administrator. (ม.ป.ป.). *MySQL คืออะไร*. เข้าถึงได้จาก SVGROUP: <http://www.svg-web.net/blog/detail/1#>
- Chakit Crengtummakit. (8 พฤษภาคม 2018). *Data Structure คืออะไร*. เข้าถึงได้จาก GlurGeek: <https://www.glurgeek.com/education/data-structure-คืออะไร/>
- Laem. (11 ตุลาคม 2011). *.NET Framework คืออะไร มีที่มาและความสำคัญอย่างไร*. เข้าถึงได้จาก notebookspec: <https://notebookspec.com/web/88056-net-framework-คืออะไร-ที่มาและความ>
- Muhammad Raza. (29 August 2018). *bmc blogs*. เข้าถึงได้จาก DBMS: An Intro to Database Management Systems: <https://www.bmc.com/blogs/dbms-database-management-systems/>
- phpMyAdmin คืออะไร*. (14 มีนาคม 2560). เข้าถึงได้จาก mindphp: <https://www.mindphp.com/คู่มือ/73คืออะไร/2285-phpmyadmin-คืออะไร.ทส>
- UX / UI คืออะไร?* (29 พฤษภาคม 2019). เข้าถึงได้จาก DEMETER ICT: <https://www.dmit.co.th/th/ข่าวสาร/ux-vs-ui/>
- Visual Control คืออะไร*. (19 กันยายน 2015). เข้าถึงได้จาก akarapol1901: <https://akarapol1901.wordpress.com/2015/09/19/visual-control-คืออะไร>
- Visual Studio วิชวลสตูดิโอ คืออะไร*. (1 กรกฎาคม 2559). เข้าถึงได้จาก mindphp: <https://www.mindphp.com/คู่มือ/73-คืออะไร/3639-visual-studio-วิชวลสตูดิโอ-คืออะไร.html>
- แนวคิดเชิงวัตถุ*. (ม.ป.ป.). เข้าถึงได้จาก Pmtech39012003: <https://sites.google.com/site/pmtech39012003/naewkhid-cheing-watthu>

การสร้างโมเดลความสัมพันธ์ระหว่างข้อมูล: ER-DIAGRAM. (ม.ป.ป.). เข้าถึงได้จาก learningsystem:

<http://learningsystem.6te.net/?page=10>

กาญจพงศ์ ปัญญาดี. (ม.ป.ป.). AppServ คืออะไร? เข้าถึงได้จาก AppServ : Apache + PHP +

MYSQL: <https://www.appserv.org/th/เกี่ยวกับ/>

ภาษา C#. (11 กุมภาพันธ์ 2016). เข้าถึงได้จาก marcuscode:

<http://marcuscode.com/lang/csharp>

องค์ประกอบของระบบฐานข้อมูล(Database). (2 กรกฎาคม 2555). เข้าถึงได้จาก Geography 29:

<http://yenchamm.blogspot.com/2012/07/database.html>

ภาคผนวก

โค้ดทั้งหมดของผู้วิจัยนี้

1.โค้ดหน้าเข้าสู่ระบบ

```
using MySql.Data.MySqlClient;
using System;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media.Imaging;
namespace Datadisplay_Rev4
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class Loginwindow : Window
    {
        static public Window lg;
        public string conString
        ="host=localhost;user=root;password=12345678;database=test_db";
        public Loginwindow()
        {
            InitializeComponent();
            SQLCore.Conn = new MySqlConnection(conString);
            lg = this;
        }
        private void close_Click(object sender, RoutedEventArgs e)
        {
            this.Close();
        }

        private void Signin_Click(object sender, RoutedEventArgs e)
```

```

{
try
{
SQLCore.Conn.Open();//เปิดข้อมูลขึ้นมา
SQLCore.cmdText = "SELECT * From test_db.tb_userlist WHERE username = " +
usernameBox.Text.Trim() + " And password = " + passwordBox.Password.Trim() + """;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
int numRows = Convert.ToInt32(SQLCore.Cmd.ExecuteScalar());
if (numRows > 0)
{
SQLCore.cmdText = "Select * from tb_userlist where username = " +
usernameBox.Text.Trim() + """;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
SQLCore.dr = SQLCore.Cmd.ExecuteReader();
if (SQLCore.dr != null)
{
SQLCore.dr.Read();
int id = (int)SQLCore.dr["id"];
Login.Status = (string)SQLCore.dr["status"];
Login.Username = (string)SQLCore.dr["firstname"];
SQLCore.dr.Close();
DateTime TT = DateTime.Now;
string format = "yyyy-MM-dd HH:mm:ss";
SQLCore.cmdText = "update test_db.tb_userlist set timestamp= " +
TT.ToString(format) + " where id = " + id.ToString();
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
SQLCore.Cmd.ExecuteNonQuery();
SQLCore.Conn.Close();
MainUserWindow Mus = new MainUserWindow();
Mus.Show();
//MainWindow Mwd = new MainWindow();

```

```

//Mwd.Show();
this.Close();
}
}
else
{
//MessageBox.Show("กรอกข้อมูลผิดพลาด");
MessageBox.Show("Cannot login the system because Username or password is
incorrect !!!", "Alert!!", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
SQLCore.Conn.Close();
}
catch (Exception ex)
{
MessageBox.Show("เกิดข้อผิดพลาด: " + ex.Message);
}
}
public Image LoadImage(string img)
{
byte[] bytes = Convert.FromBase64String(img);
Image image = new Image();
BitmapImage myBitmapImage = new BitmapImage();
using (MemoryStream ms = new MemoryStream(bytes))
{
image.Source = BitmapFrame.Create(ms, BitmapCreateOptions.IgnoreImageCache,
BitmapCacheOption.OnLoad); ;
}
return image;
}
}
}

```

2. โค้ดหน้าการทำงานหลัก

```

using Microsoft.Win32;
using MySql.Data.MySqlClient;
using System;
using System.Windows;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using EExcel= Microsoft.Office.Interop.Excel;
using wdf = System.Windows.Forms;

namespace Datadisplay_Rev4
{
    /// <summary>
    /// Interaction logic for MainUserWindow.xaml
    /// </summary>
    public partial class MainUserWindow : Window
    {
        static public PageDataInputUser MP = new PageDataInputUser();
        static public MainWindow Mwd = new MainWindow();
        private bool stateMaximiz = false;
        public MainUserWindow()
        {
            InitializeComponent();

            PageDataInputUser.PD1.StateLine = "ARQ";
            PageDataInputUser.PD1.Inint();
            PageDataInputUser.PD1.initConfic();

            PageDataInputUser.PD2.StateLine = "SubAssy";

```

```

PageDataInputUser.PD2.Inint();
PageDataInputUser.PD2.initConfiC();

```

```

PageDataInputUser.PD3.StateLine = "SKD";
PageDataInputUser.PD3.Inint();
PageDataInputUser.PD3.initConfiC();

```

```

PageDataInputUser.PDH1.StateLineInput = "ARQ";
PageDataInputUser.PDH1.Inint();

```

```

PageDataInputUser.PDH2.StateLineInput = "SubAssy";
PageDataInputUser.PDH2.Inint();

```

```

PageDataInputUser.PDH3.StateLineInput = "SKD";
PageDataInputUser.PDH3.Inint();

```

```

FrameData.Content = MP;
}

```

```

private void menubar_MouseDown(object sender, MouseButtonEventArgs e)
{
    //DragMove();
}

```

```

private void CCloseMu_Click(object sender, RoutedEventArgs e)
{
    Application.Current.Shutdown();
}

```

```

private void Export_Click(object sender, RoutedEventArgs e)
{
    SaveFileDialog sfd = new SaveFileDialog()
    {
        Title = "เลือกบันทึกไฟล์",
        Filter = "Excel Workbook|*.xls",
        ValidateNames = true
    };

    if (sfd.ShowDialog() == true)
    {
        EExcel.Application excelApp = new EExcel.Application();
        if (excelApp != null)
        {
            EExcel.Workbook excelWorkbook = excelApp.Workbooks.Add();
            //EExcel.Worksheet excelWorksheet = (EExcel.Worksheet)excelWorkbook.Sheets.Add();
            EExcel.Worksheet excelWorksheet = (EExcel.Worksheet)excelWorkbook.Sheets[1];
            //ws = wb.Worksheets[Sheet];
            excelWorksheet.Cells[1, 1] = "No";
            excelWorksheet.Cells[1, 2] = "Date";
            excelWorksheet.Cells[1, 3] = "Time";
            excelWorksheet.Cells[1, 4] = "Model";
            excelWorksheet.Cells[1, 5] = "Station";
            excelWorksheet.Cells[1, 6] = "Lot/Batch";
            excelWorksheet.Cells[1, 7] = "Vin No";
            excelWorksheet.Cells[1, 8] = "MFG";
            excelWorksheet.Cells[1, 9] = "Plan";
            excelWorksheet.Cells[1, 10] = "Actual";
            excelWorksheet.Cells[1, 11] = "TakeTime";
            excelWorksheet.Cells[1, 12] = "Status";

```

```

SQLCore.Conn.Open();
SQLCore.cmdText = "Select * from test_db.tb_historydata where state = " + "ARQ"+
"";
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
MySQLDataReader reader = SQLCore.Cmd.ExecuteReader();
int countRow = 2;
while (reader.Read())
{
    excelWorksheet.Cells[countRow, 1] = Convert.ToString(countRow-1);
    excelWorksheet.Cells[countRow, 2] = (string)reader["Date"];
    excelWorksheet.Cells[countRow, 3] = (string)reader["Time"];
    excelWorksheet.Cells[countRow, 4] = (string)reader["Model"];
    excelWorksheet.Cells[countRow, 5] = (string)reader["Strn"];
    excelWorksheet.Cells[countRow, 6] = (string)reader["Lot_Batch"];
    excelWorksheet.Cells[countRow, 7] = (string)reader["Vin_No"];
    excelWorksheet.Cells[countRow, 8] = (string)reader["MFG"];
    excelWorksheet.Cells[countRow, 9] = (string)reader["Plan"];
    excelWorksheet.Cells[countRow, 10] = (string)reader["Actual"];
    excelWorksheet.Cells[countRow, 11] = (string)reader["TakeTime"];
    excelWorksheet.Cells[countRow, 12] = (string)reader["Status"];

    //excelApp.ActiveWorkbook.Save();
    countRow++;
}
SQLCore.Conn.Close();
excelApp.ActiveWorkbook.SaveAs(sfd.FileName,
Excel.XlFileFormat.xlWorkbookNormal);
excelWorkbook.Close();
excelApp.Quit();

```

```

}
}
}

```

```
private void DISPLAY2_Click(object sender, RoutedEventArgs e)
```

```

{
    if (wdf.Screen.AllScreens.Length > 1)
    {
        wdf.Screen s = wdf.Screen.AllScreens[1];
        System.Drawing.Rectangle r = s.WorkingArea;
        Mwd.Top = r.Top;
        Mwd.Left = r.Left;
        Mwd.Show();
        Mwd.FullScreen_Click(null, null);

```

```

    }
    else
    {
        Mwd.Show();
        Mwd.FullScreen_Click(null, null);
    }
}

```

```
private void CCloseMu_MouseEnter(object sender, MouseEventArgs e)
```

```

{
    CCloseMu.Background = new SolidColorBrush(Color.FromArgb(255, 238, 32, 32)); //Red
}

```

```
private void CCloseMu_MouseLeave(object sender, MouseEventArgs e)
```

```

{
    CCloseMu.Background = new SolidColorBrush(Colors.Transparent);
}

```

```
}
```

```
private void MinimizMu_Click(object sender, RoutedEventArgs e)
{
    this.WindowState = WindowState.Minimized;
}
```

```
private void MaximizMu_Click(object sender, RoutedEventArgs e)
{

    if (stateMaximiz == true)
    {
        this.WindowState = WindowState.Maximized;
        ImageMM.Source = new BitmapImage(new Uri(@"image/restore_down_15px.png",
        UriKind.Relative));
        ImageMM.Height = 11;
        ImageMM.Width = 13;
        stateMaximiz = false;
    }
    else
    {
        this.WindowState = WindowState.Normal;
        ImageMM.Source = new BitmapImage(new Uri(@"image/maximize_button_32px.png",
        UriKind.Relative));
        ImageMM.Height = 15;
        ImageMM.Width = 18;
        stateMaximiz = true;
    }

}
```

```
private void MaximizMu_MouseEnter(object sender, MouseEventArgs e)
```

```

{
    MaximizMu.Background = new SolidColorBrush(Color.FromArgb(255, 238, 32,
    32)); //Red
}

private void MaximizMu_MouseLeave(object sender, MouseEventArgs e)
{
    MaximizMu.Background = new SolidColorBrush(Colors.Transparent);
}

private void MinimizMu_MouseEnter(object sender, MouseEventArgs e)
{
    MinimizMu.Background = new SolidColorBrush(Color.FromArgb(255, 238, 32,
    32)); //Red
}

private void MinimizMu_MouseLeave(object sender, MouseEventArgs e)
{
    MinimizMu.Background = new SolidColorBrush(Colors.Transparent);
}

private void gg_MouseDown(object sender, MouseButtonEventArgs e)
{
    //DragMove();
}

private void SetTime_Click(object sender, RoutedEventArgs e)
{
    settimebleak stb = new settimebleak();
    stb.Show();
}

```

```
private void Datainput_Click(object sender, RoutedEventArgs e)
{
    FrameData.Content = MP;
}

private void AccountSet_Click(object sender, RoutedEventArgs e)
{
    AddAccountsWindow ac = new AddAccountsWindow();
    ac.Show();
}
}
```

3. โค้ดหน้าป้อนข้อมูลการผลิตรายงาน

```

using Microsoft.Win32;
using MySql.Data.MySqlClient;
using System;
using System.Collections;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Threading;
using EExcel = Microsoft.Office.Interop.Excel;
using wdf = System.Windows.Forms;

namespace Datadisplay_Rev4
{
    /// <summary>
    /// Interaction logic for PageDataARQ.xaml
    /// </summary>

```

```

public partial class PageDataARQ : Page
{
    List<TodoItemDt> items = new List<TodoItemDt>();
    List<string> items_message = new List<string>();
    private bool ButtonUpdate = false;
    public string StateLineInput;
    public string StateLine;
    private DispatcherTimer Timer;
    public DispatcherTimer TimerCountDown;
    public int minute = 59;
    private Double Int_TakeTime = 59;
    private int time = 60;
    public int id;
    public int St_ID;
    bool StateVin = false;
    bool StateBatch = false;
    int countActule = 0;
    bool StateModeQrcode = false;
    public PageDataARQ()
    {
        InitializeComponent();
        Timer = new DispatcherTimer();
        Timer.Interval = new TimeSpan(0, 0, 1);

        Timer.Tick += timetikeOT;
        Timer.Tick += Check;
        Timer.Start();

        TimerCountDown = new DispatcherTimer();
        TimerCountDown.Interval = new TimeSpan(0, 0, 1);
        TimerCountDown.Tick += Timer_Tick;
    }
}

```

```

TxtScanner.Focus();
confirm.Visibility = Visibility.Hidden;
}

```

```

public class TodoItemDt
{
    public string No { get; set; }
    public string Date { get; set; }
    public string Time { get; set; }
    public string Model { get; set; }
    public string Stn { get; set; }
    public string Lot_Batch { get; set; }
    public string Vin_No { get; set; }
    public string MFG_NO { get; set; }
    public string Plan { get; set; }
    public string Actual { get; set; }
    public string TakeTime { get; set; }
    public string Status { get; set; }
    public int NoSQL { get; set; }
}

void Check(object sender, EventArgs e) // ฟังก์ชันการเช็คเปิดโหมด QR code อยู่หรือไม่
{
    if (StateModeQrcode == true)
    {
        if (TxtScanner.Text == "" || String.IsNullOrEmpty(TxtScanner.Text)==true)
        {
            TxtScanner.Focus();
        }
    }
    SM_MODEL.Content = StateLine;
}

```

```

void Timer_Tick(object sender, EventArgs e)
{
    if (minute <= 0 && time <= 0)
    {
        minute = Convert.ToInt32(Int_TakeTime) + 1;
        Console.WriteLine("IIII : " + minute.ToString());
    }
    else if (time == 0)
    {
        minute--;
        time = 60;
    }
    else
    {
        time--;
        SM_COUNTDOWN.Content = minute.ToString("D2") + ":" + (time % 60).ToString("D2");
        ARC1.EndAngle = Convert.ToDouble(minute / Int_TakeTime) * 360;
    }
}

public void Inint()
{
    for (int i = items.Count - 1; i >= 0; i--)
    {
        items.RemoveAt(i);
    }
    SQLCore.Conn.Open();
    SQLCore.cmdText = "Select * from test_db.tb_historydata where state = '" +
        StateLine + "'";
    SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
    MySqlDataReader reader = SQLCore.Cmd.ExecuteReader();

```

```

int countRow = 1;
while (reader.Read())
{
    DateTime _date = DateTime.Now;
    items.Add(new TodoItemDt()
    {
        No = countRow.ToString(),
        Date = (string)reader["Date"],
        Time = (string)reader["Time"],
        Model = (string)reader["Model"],
        Stn = (string)reader["Stn"],
        Lot_Batch = (string)reader["Lot_Batch"],
        Vin_No = (string)reader["Vin_No"],
        MFG_NO = (string)reader["MFG"],
        Plan = (string)reader["Plan"],
        Actual = (string)reader["Actual"],
        TakeTime = (string)reader["TakeTime"],
        Status = (string)reader["Status"],
        NoSQL = (int)reader["No"]
    });
    countRow++;
}
SQLCore.Conn.Close();
DataListDt.ItemsSource = null;
DataListDt.ItemsSource = items;
}

bool flagtime = false;

bool ss_MFG = false;
bool ss_VIN = false;
bool ss_MFG100 = false;

```

```

private void Insert_Click(object sender, RoutedEventArgs e)
{
    if (!String.IsNullOrEmpty(MFG.Text) && Regex.IsMatch(MFG.Text, "[0-9]*$") &&
        MFG.Text.Length <= 4 && !String.IsNullOrEmpty(VIN.Text) && Regex.IsMatch(VIN.Text,
            "[a-zA-Z0-9]*$") && VIN.Text.Length <= 25 && VIN.Text.Length >= 7)
    {
        #region main insert
        Insert.IsEnabled = false;
        if (!flagtime)
        {
            TimerCountDown.Start();
            flagtime = true;
        }
        DataListDt.ScrollIntoView(DataListDt.Items[DataListDt.Items.Count - 1]);
        SM_COUNTDOWN.Visibility = Visibility.Visible;
        try
        {
            if (!ButtonUpdate)
            {
                confirm.Visibility = Visibility.Visible;
            }
        }
        else
        {
            Insert.IsEnabled = true;
        }
        time = 59;
        SM_ACTUAL.Content = ACTUAL.Text;
        SM_PLAN.Content = PLAN.Text.Trim();
        SM_DIFF.Content = Convert.ToInt32(ACTUAL.Text) - Convert.ToInt32(PLAN.Text);
    }
}

```

```
SM_TAKTTIME.Content = Convert.ToString(505 / Convert.ToInt32(PLAN.Text.Trim()));
```

```
if (StateModeQrcode)
```

```
{
```

```
SM_MFG.Content = MFG.Text.Trim();
```

```
}
```

```
else
```

```
{
```

```
if (!ButtonUpdate)
```

```
{
```

```
MFG.Text = MFG.Text + "/50";
```

```
SM_MFG.Content = MFG.Text;
```

```
}
```

```
}
```

```
SM_MODEL.Content = MODEL.Text.Trim();
```

```
SM_LOT.Content = LOT.Text.Trim();
```

```
SM_STNL.Content = STN.Text.Trim();
```

```
SM_VIN.Content = VIN.Text.Trim();
```

```
minute = Convert.ToInt32(505 / Convert.ToInt32(PLAN.Text.Trim()) - 1);
```

```
Int_TakeTime = Convert.ToDouble(SM_TAKTTIME.Content) - 1;
```

```
if (StateLineInput == "ARQ")
```

```
{
```

```
#region paren page1 ARQ
```

```
MainWindow.P1_ARQ.grid_ed.Visibility = Visibility.Visible;
```

```
MainWindow.P1_ARQ.TBCountDown.Visibility = Visibility.Visible;
```

```
MainWindow.P1_ARQ.Input_Actual.Content = ACTUAL.Text;
```

```
MainWindow.P1_ARQ.Input_Actual.Content = ACTUAL.Text.Trim();
```

```
MainWindow.P1_ARQ.Input_Plane.Content = PLAN.Text.Trim();
```

```

MainWindow.P1_ARQ.Input_Diff.Content = Convert.ToInt32(ACTUAL.Text) -
Convert.ToInt32(PLAN.Text);

MainWindow.P1_ARQ.Input_TakeTime.Content = SM_TAKTTIME.Content;
MainWindow.P1_ARQ.Int_TakeTime = Convert.ToDouble(minute);
MainWindow.P1_ARQ.minute = Convert.ToInt32(minute);


MainWindow.P1_ARQ.Input_Textwork.Content = OT_NT.Content;
MainWindow.P1_ARQ.Input_MFG.Content = MFG.Text;
MainWindow.P1_ARQ.Input_Model.Content = MODEL.Text;
MainWindow.P1_ARQ.Input_Lot.Content = LOT.Text;
MainWindow.P1_ARQ.Input_Stn.Content = STN.Text;
MainWindow.P1_ARQ.Input_Vin.Content = VIN.Text;
//MainWindow.P1_ARQ.Input_CountCar.Content = TxtCOUNTCAR.Text;//ถ้าสุด

#endregion
}

else if (StateLineInput == "SubAssy")
{

#region paren page1 SubAssy
MainWindow.P1_SubAssy.grid_ed.Visibility = Visibility.Visible;
MainWindow.P1_SubAssy.TBCountDown.Visibility = Visibility.Visible;
MainWindow.P1_SubAssy.Input_Actual.Content = ACTUAL.Text.Trim();
MainWindow.P1_SubAssy.Input_Plane.Content = PLAN.Text.Trim();
MainWindow.P1_SubAssy.Input_Diff.Content = Convert.ToInt32(ACTUAL.Text) -
Convert.ToInt32(PLAN.Text);

MainWindow.P1_SubAssy.Input_TakeTime.Content = SM_TAKTTIME.Content;
MainWindow.P1_SubAssy.Int_TakeTime = Convert.ToDouble(minute);
MainWindow.P1_SubAssy.minute = Convert.ToInt32(minute);

```

```

MainWindow.P1_SubAssy.Input_Textwork.Content = OT_NT.Content;
MainWindow.P1_SubAssy.Input_MFG.Content = MFG.Text;
MainWindow.P1_SubAssy.Input_Model.Content = MODEL.Text;
MainWindow.P1_SubAssy.Input_Lot.Content = LOT.Text;
MainWindow.P1_SubAssy.Input_Stn.Content = STN.Text;
MainWindow.P1_SubAssy.Input_Vin.Content = VIN.Text;
#endregion
}
else if (StateLineInput == "SKD")
{
#region paren page1 SKD
MainWindow.P1_SKD.grid_ed.Visibility = Visibility.Visible;
MainWindow.P1_SKD.TBCountDown.Visibility = Visibility.Visible;
MainWindow.P1_SKD.Input_Actual.Content = ACTUAL.Text.Trim();
MainWindow.P1_SKD.Input_Plane.Content = PLAN.Text.Trim();
MainWindow.P1_SKD.Input_Diff.Content = Convert.ToInt32(ACTUAL.Text) -
Convert.ToInt32(PLAN.Text);
MainWindow.P1_SKD.Input_TakeTime.Content = SM_TAKTTIME.Content;
MainWindow.P1_SKD.Int_TakeTime = Convert.ToDouble(minute);
MainWindow.P1_SKD.minute = Convert.ToInt32(minute);
MainWindow.P1_SKD.Input_Textwork.Content = OT_NT.Content;
MainWindow.P1_SKD.Input_MFG.Content = MFG.Text;
MainWindow.P1_SKD.Input_Model.Content = MODEL.Text;
MainWindow.P1_SKD.Input_Lot.Content = LOT.Text;
MainWindow.P1_SKD.Input_Stn.Content = STN.Text;
MainWindow.P1_SKD.Input_Vin.Content = VIN.Text;
#endregion
}
//ในส่วนของหน้าจออีกหน้าจอที่ขึ้นTV
DateTime _date = DateTime.Now;

```

```

SQLCore.Conn.Open();
if (ButtomUpdate == true)
{
    SQLCore.cmdText = SQLCore.cmdText = "UPDATE test_db.tb_historydata SET Model=
    " + MODEL.Text.Trim() + ", Stn= " + STN.Text.Trim() + ",Actual = " + ACTUAL.Text + "
    ,Plan = " + PLAN.Text + " ,Lot_Batch = " + LOT.Text + ",Vin_No = " + VIN.Text +
    ",MFG = " + MFG.Text + ",TakeTime = " + TAKETIME.Text + " WHERE No = " + St_ID;
    SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
    SQLCore.Cmd.ExecuteNonQuery();
}
else
{
    SQLCore.cmdText = "INSERT INTO test_db.tb_historydata
    (State,Date,Time,Model,Stn,Lot_Batch,Vin_No,MFG,Plan,Actual,TakeTime,Status)
    VALUES
    (?St,?Date,?Time,?Model,?Stn,?Lot_Batch,?Vin_No,?MFG,?Plan,?Actual,?Take Time,?Statu
    s)";
    SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
    SQLCore.Cmd.Parameters.AddWithValue("?St", StateLine);
    SQLCore.Cmd.Parameters.AddWithValue("?Date", _date.ToString("dd/MMM/yy"));
    SQLCore.Cmd.Parameters.AddWithValue("?Time", _date.ToString("hh:mm tt"));
    SQLCore.Cmd.Parameters.AddWithValue("@?Model", MODEL.Text.Trim());
    SQLCore.Cmd.Parameters.AddWithValue("?Stn", STN.Text.Trim());
    SQLCore.Cmd.Parameters.AddWithValue("?Lot_Batch", LOT.Text.Trim());
    SQLCore.Cmd.Parameters.AddWithValue("?Vin_No", VIN.Text.Trim());
    SQLCore.Cmd.Parameters.AddWithValue("?MFG", MFG.Text.Trim());
    SQLCore.Cmd.Parameters.AddWithValue("?Plan", PLAN.Text.Trim());
    SQLCore.Cmd.Parameters.AddWithValue("?Actual", ACTUAL.Text.Trim());
    SQLCore.Cmd.Parameters.AddWithValue("?TakeTime", TAKETIME.Text.Trim());
    SQLCore.Cmd.Parameters.AddWithValue("?Status", "RUNNING");
    SQLCore.Cmd.ExecuteNonQuery();
}

```

```

WindowScaneCuccess wsc = new WindowScaneCuccess();
if (wdf.Screen.AllScreens.Length > 1)
{
    Soundsuccess.Stop();
    wdf.Screen s = wdf.Screen.AllScreens[1];
    wsc.Top = s.WorkingArea.Top;
    wsc.Left = s.WorkingArea.Left;
    wsc.Show();
    wsc.WindowState = WindowState.Maximized;
    Soundsuccess.Play();

}
else
{
    Soundsuccess.Stop();
    wdf.Screen s = wdf.Screen.AllScreens[0];
    wsc.Top = s.WorkingArea.Top;
    wsc.Left = s.WorkingArea.Left;
    wsc.Show();
    wsc.WindowState = WindowState.Maximized;
    Soundsuccess.Play();

}
}
SQLCore.Conn.Close();
Inint();
ClearAll();
}
catch (Exception)
{

```

```

MessageBox.Show("กรุณกรอกข้อมูลให้ครบ");
}
#endregion
}
else
{
if (String.IsNullOrEmpty(MFG.Text))
{
MessageBox.Show("กรุณกรอกข้อมูลในช่อง MFG ");

}
else if (!Regex.IsMatch(MFG.Text, "^[0-9]*$"))
{
MessageBox.Show("ช่อง MFG สามารถกรอกได้เพียงตัวเลขเท่านั้น ");
}
else if (MFG.Text.Length > 4)
{
MessageBox.Show("ช่อง MFG สามารถกรอกสูงสุดเพียง 4 หลัก ");
}
else if (String.IsNullOrEmpty(VIN.Text))
{
MessageBox.Show("กรุณกรอกข้อมูลในช่อง VIN ");
}
else if (!Regex.IsMatch(VIN.Text, "^[a-zA-Z0-9]*$"))
{
MessageBox.Show("ช่อง VIN ไม่อนุญาตให้กรอกอักขระพิเศษ ");
}
else if (VIN.Text.Length > 25)
{
MessageBox.Show("ช่อง VIN สามารถกรอกสูงสุดเพียง 25 หลัก ");
}
}

```

```

}
else if (VIN.Text.Length < 7)
{
    MessageBox.Show("ช่อง VIN สามารถกรอกได้สูงสุดเพียง 7 หลัก ");
}
else
{
}
}
}

private void Delete_Click(object sender, RoutedEventArgs e)
{
    St_ID = (items[DataListDt.SelectedIndex].NoSQL);
    SQLCore.Conn.Open();
    SQLCore.cmdText = SQLCore.cmdText = "DELETE FROM test_db.tb_historydata
    WHERE No =" + St_ID.ToString();
    SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
    SQLCore.Cmd.ExecuteNonQuery();
    SQLCore.Conn.Close();
    Inint();
}

private void Edit_Click(object sender, RoutedEventArgs e)
{
    try
    {
        Insert.IsEnabled = true;
        Insert.Content = "Update";
        ACTUAL.Text = (items[DataListDt.SelectedIndex].Actual);
        PLAN.Text = (items[DataListDt.SelectedIndex].Plan);
        TAKETIME.Text = (items[DataListDt.SelectedIndex].TakeTime);
    }
}

```

```

MFG.Text = (items[DataListDt.SelectedIndex].MFG_NO);
MODEL.Text = (items[DataListDt.SelectedIndex].Model);
LOT.Text = (items[DataListDt.SelectedIndex].Lot_Batch);
STN.Text = (items[DataListDt.SelectedIndex].Stn);
VIN.Text = (items[DataListDt.SelectedIndex].Vin_No);
St_ID = (items[DataListDt.SelectedIndex].NoSQL);
ButtomUpdate = true;
Delete.IsEnabled = false;
Edit.IsEnabled = false;
MFG.IsEnabled = true;
VIN.IsEnabled = true;
}
catch (Exception)
{
    MessageBox.Show("ไม่พบข้อมูลให้แก้ไข");
}
}

private void ClearAll()
{
    Insert.Content = "Insert";
    MFG.Text = "";
    VIN.Text = "";
    ButtomUpdate = false;
    Delete.IsEnabled = true;
    Edit.IsEnabled = true;
}

private void Cancel_Click(object sender, RoutedEventArgs e)
{
    ClearAll();
}

void timetikeOT(object sender, EventArgs e)

```

```

{
    TimeSpan start = new TimeSpan(8,00,0); //10 o'clock
    TimeSpan end = new TimeSpan(17,45,0); //12 o'clock
    TimeSpan now = DateTime.Now.TimeOfDay;

    if ((now > start) && (now < end))
    {
        OT_NT.Content = "NT";
    }
    else
    {
        OT_NT.Content = "OT";
    }
}

private void Export_Click(object sender, RoutedEventArgs e)
{
    SaveFileDialog sfd = new SaveFileDialog()
    {
        Title = "เลือกบันทึกไฟล์",
        Filter = "Excel Workbook|*.xls",
        ValidateNames = true
    };

    if (sfd.ShowDialog() == true)
    {
        EExcel.Application excelApp = new EExcel.Application();
        if (excelApp != null)
        {
            EExcel.Workbook excelWorkbook = excelApp.Workbooks.Add();
            EExcel.Worksheet excelWorksheet = (EExcel.Worksheet)excelWorkbook.Sheets[1];

```

```

excelWorksheet.Cells[1, 1] = "No";
excelWorksheet.Cells[1, 2] = "Date";
excelWorksheet.Cells[1, 3] = "Time Start";
excelWorksheet.Cells[1, 4] = "Time Finish";
excelWorksheet.Cells[1, 5] = "Model";
excelWorksheet.Cells[1, 6] = "Station";
excelWorksheet.Cells[1, 7] = "Lot/Batch";
excelWorksheet.Cells[1, 8] = "Vin No";
excelWorksheet.Cells[1, 9] = "MFG";
excelWorksheet.Cells[1, 10] = "Plan";
excelWorksheet.Cells[1, 11] = "Actual";
excelWorksheet.Cells[1, 12] = "TakeTime";
excelWorksheet.Cells[1, 13] = "Status";

```

```

SQLCore.Conn.Open();
SQLCore.cmdText = "Select * from test_db.tb_historydata where state = " +
StateLine + """;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
MySqlDataReader reader = SQLCore.Cmd.ExecuteReader();
int countRow = 2;
while (reader.Read())
{
    excelWorksheet.Cells[countRow, 1] = Convert.ToString(countRow - 1);
    excelWorksheet.Cells[countRow, 2] = (string)reader["Date"];
    excelWorksheet.Cells[countRow, 3] = (string)reader["Time"];
    excelWorksheet.Cells[countRow, 4] = (string)reader["TimeFinish"];
    excelWorksheet.Cells[countRow, 5] = (string)reader["Model"];
    excelWorksheet.Cells[countRow, 6] = (string)reader["Stn"];
    excelWorksheet.Cells[countRow, 7] = (string)reader["Lot_Batch"];
}

```

```

excelWorksheet.Cells[countRow, 8] = (string)reader["Vin_No"];
excelWorksheet.Cells[countRow, 9] = (string)reader["MFG"];
excelWorksheet.Cells[countRow, 10] = (string)reader["Plan"];
excelWorksheet.Cells[countRow, 11] = (string)reader["Actual"];
excelWorksheet.Cells[countRow, 12] = (string)reader["TakeTime"];
excelWorksheet.Cells[countRow, 13] = (string)reader["Status"];
countRow++;
}
SQLCore.Conn.Close();
excelApp.ActiveWorkbook.SaveAs(sfd.FileName,
EExcel.XlFileFormat.xlWorkbookNormal);
excelWorkbook.Close();
excelApp.Quit();
}
}
}

```

```

private void TxtScanner_KeyUp(object sender, KeyEventArgs e)
{
    string[] str;
    string textDT;
    int Lengthstring;
    if (e.Key == Key.Enter)
    {
        TextBox txtBox = e.Source as TextBox;
        if (txtBox != null)
        {
            this.TxtScanner.Text = txtBox.Text;
            textDT = txtBox.Text;
            Lengthstring = textDT.Length;
            str = textDT.Split(",").ToCharArray();

```

```

if (Lengthstring>10)
{
VIN.Text = textDT;
StateVin = true;
}
else
{
this.TxtScanner.Background = new SolidColorBrush(Colors.LightGray);
MFG.Text = Convert.ToString(str[0]).Trim();
LOT.Text = Convert.ToString(str[1]).Trim();
StateBatch = true;
}
}
TxtScanner.Text = "";
if (StateVin == true && StateBatch == true)
{
Insert_Click(null, null);
StateBatch = false;
StateVin = false;
}
}
}

private void ToggleButton_Checked(object sender, RoutedEventArgs e)
{
if (String.IsNullOrEmpty(ACTUAL.Text))
{
MessageBox.Show("กรุณากดรอกข้อมูลในช่อง ACTUAL ");
CKBox.IsChecked = false;
}
}

```

```

else if (String.IsNullOrEmpty(PLAN.Text))
{
    MessageBox.Show("กรุณกรอกข้อมูลในช่อง PLAN ");
    CKBox.IsChecked = false;
}

else if (String.IsNullOrEmpty(TAKETIME.Text))
{
    MessageBox.Show("กรุณกรอกข้อมูลในช่อง TAKETIME ");
    CKBox.IsChecked = false;
}
else
{
    statusQR.Content = "ON";
    statusQR.Foreground = new SolidColorBrush(Color.FromArgb(255, 65, 230,
39)); //green
    .Foreground = new SolidColorBrush(Color.FromArgb(255, 255, 0,
0)); //green
    ACTUAL.IsEnabled = false;
    PLAN.IsEnabled = false;
    TAKETIME.IsEnabled = false;
    LOT.IsEnabled = false;
    VIN.IsEnabled = false;
    MFG.IsEnabled = false;
    StateModeQrcode = true;
    TxtScanner.Focus();
}

}

private void CKBox_Unchecked(object sender, RoutedEventArgs e)
{

```

```

statusQR.Content = "OFF";
statusQR.Foreground = new SolidColorBrush(Color.FromArgb(255, 255, 0, 0));
MFG.IsEnabled = true;
VIN.IsEnabled = true;
StateModeQrcode = false;

}

```

```

string plan;
string actual;
string taketime;
string Batch;
string Stn;
string Day;
string planOT;
public void initConfc()// confic ค่าต่างๆ
{
    string LastDay = (items[DataListDt.Items.Count - 1].Date);
    int No = Convert.ToInt32(items[DataListDt.Items.Count - 1].NoSQL);
    DateTime DayNow = DateTime.Now;
    int Rowstate = (StateLine == "ARQ") ? 1 : (StateLine == "SubAssy") ? 2 : 3;
    SQLCore.Conn.Open();
    SQLCore.cmdText = "Select * from test_db.tb_count where id = " + Rowstate + "";
    SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
    SQLCore.dr = SQLCore.Cmd.ExecuteReader();
    if (SQLCore.dr != null)
    {
        SQLCore.dr.Read();
        plan = (string)SQLCore.dr["Plan"];
        actual = (string)SQLCore.dr["Actual"];
        taketime = (string)SQLCore.dr["TakeTime"];
    }
}

```

```

Batch = (string)SQLCore.dr["Batch"];
Stn = (string)SQLCore.dr["Stn"];
Day = (string)SQLCore.dr["Day"];
planOT = (string)SQLCore.dr["PlanOT"];
SQLCore.dr.Close();
}
SQLCore.Conn.Close();
Console.WriteLine("DayNow : " + DayNow.ToString("dd/MMM/yy") + " Day : " + LastDay
+ " State : " + StateLine + "NO :"+ No.ToString());
if (Convert.ToString(DayNow.ToString("dd/MMM/yy")) != LastDay)
{
Console.WriteLine("reset");
ACTUAL.Text = "0";
SQLCore.Conn.Open();
SQLCore.cmdText = SQLCore.cmdText = "UPDATE test_db.tb_count SET Actual= " +
ACTUAL.Text + " WHERE id = " + Rowstate;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
SQLCore.Cmd.ExecuteNonQuery();
SQLCore.Conn.Close();
}
else
{
ACTUAL.Text = actual;
}
PLAN.Text = plan;
TAKETIME.Text = taketime;
LOT.Text = Batch;
STN.Text = Stn;

}

```

```

public void updatecountcar()
{

}

```

```

public void updateActual()
{
int Rowstate = (StateLine == "ARQ") ? 1 : (StateLine == "SubAssy") ? 2 : 3;
countActule = Convert.ToInt32(ACTUAL.Text);
countActule++;
SQLCore.Conn.Open();
SQLCore.cmdText = SQLCore.cmdText = "UPDATE test_db.tb_count SET Actual= " +
countActule.ToString() + " WHERE id = " + Rowstate;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
SQLCore.Cmd.ExecuteNonQuery();
SQLCore.Conn.Close();
ACTUAL.Text = Convert.ToString(countActule);
ACTUAL.UpdateLayout();
}

```

```

private void BT_SetACTUAL_Click(object sender, RoutedEventArgs e)
{
int Rowstate = (StateLine == "ARQ") ? 1 : (StateLine == "SubAssy") ? 2 : 3;
Inputbox IpBoxACTUAL = new Inputbox("Actual", Rowstate, ACTUAL, "Actual",
planOT);
IpBoxACTUAL.Show();
}

```

```

private void BT_SetPlan_Click(object sender, RoutedEventArgs e)

```

```

{
    int Rowstate = (StateLine == "ARQ") ? 1 : (StateLine == "SubAssy") ? 2 : 3;
    Inputbox IpBoxPlan = new Inputbox("Plan", Rowstate, PLAN, "Plan NT", planOT);
    IpBoxPlan.Show();
}

private void BT_SetTAKETIME_Click(object sender, RoutedEventArgs e)
{
    int Rowstate = (StateLine == "ARQ") ? 1 : (StateLine == "SubAssy") ? 2 : 3;
    Inputbox IpBoxTAKETIME = new Inputbox("TakeTime", Rowstate, TAKETIME,
    "TakeTime", planOT);
    IpBoxTAKETIME.Show();
}

private void BT_SetCOUNTCAR_Click(object sender, RoutedEventArgs e)
{
}

private void BT_SetBATCH_Click(object sender, RoutedEventArgs e)
{
    int Rowstate = (StateLine == "ARQ") ? 1 : (StateLine == "SubAssy") ? 2 : 3;
    Inputbox IpBoxCOUNTCAR = new Inputbox("Batch", Rowstate, LOT, "Batch", planOT);
    IpBoxCOUNTCAR.Show();
}

private void BT_SetStn_Click(object sender, RoutedEventArgs e)
{
    int Rowstate = (StateLine == "ARQ") ? 1 : (StateLine == "SubAssy") ? 2 : 3;
    Inputbox IpBoxStn = new Inputbox("Stn", Rowstate, STN, "STN.", planOT);
    IpBoxStn.Show();
}

```

```

}

public void startOT ()
{
    int Rowstate = (StateLine == "ARQ") ? 1 : (StateLine == "SubAssy") ? 2 : 3;
    ACTUAL.Text = "0";
    PLAN.Text = planOT;
    SQLCore.Conn.Open();
    SQLCore.cmdText = SQLCore.cmdText = "UPDATE test_db.tb_count SET stateOT= " +
    "OT" + " WHERE id = " + Rowstate;
    SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
    SQLCore.Cmd.ExecuteNonQuery();
}

```

```

private void _lineNotify(string msg, int stickerPackageID, int stickerID, string pictureUrl)
{
    try
    {
        var request = (HttpWebRequest)WebRequest.Create("https://notify-
        api.line.me/api/notify");

        var postData = string.Format("message={0}", msg);
        if (stickerPackageID > 0 && stickerID > 0)
        {
            var stickerPackageId = string.Format("stickerPackageId={0}", stickerPackageID);
            var stickerId = string.Format("stickerId={0}", stickerID);
            postData += "&" + stickerPackageId.ToString() + "&" + stickerId.ToString();
        }
        if (pictureUrl != "")
        {

```

```

var imageThumbnail = string.Format("imageThumbnail={0}", pictureUrl);
var imageFullsize = string.Format("imageFullsize={0}", pictureUrl);
postData += "&" + imageThumbnail.ToString() + "&" + imageFullsize.ToString();
}

var data = Encoding.UTF8.GetBytes(postData);

request.Method = "POST";
request.ContentType = "application/x-www-form-urlencoded";
request.ContentLength = data.Length;
request.Headers.Add("Authorization", "Bearer " + line.token);

using (var stream = request.GetRequestStream()) stream.Write(data, 0, data.Length);
var response = (HttpWebResponse)request.GetResponse();
var responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();
}
catch (Exception ex)
{
    Console.WriteLine(ex.ToString());
}
}

private void notifyPicture(string url)
{
    _lineNotify(" ", 0, 0, url);
}

private void notifySticker(int stickerID, int stickerPackageID)
{
    _lineNotify(" ", stickerPackageID, stickerID, "");
}

private void lineNotify(string msg)
{

```

```
_lineNotify(msg, 0, 0, "");
}
```

```
private void confirm_Click(object sender, RoutedEventArgs e)
{
    confirm.Visibility = Visibility.Hidden;
    Insert.IsEnabled = true;
```

```
string start = "\n" + DateTime.Now.ToString("dd-MMM-yy") + " " + StateLine + " real
time output\n" +
"report :\n" +
"\n";
```

```
string form_msg;
string Time;
string batch;
string MFG;
string Vin;
string msg = "";
```

```
updateActual();
items_message.Clear();
SQLCore.Conn.Open();//เปิดข้อมูลขึ้นมา
int No = Convert.ToInt32(items[DataListDt.Items.Count - 1].NoSQL);
SQLCore.cmdText = SQLCore.cmdText = "UPDATE test_db.tb_historydata SET
TimeFinish= " + Convert.ToString(DateTime.Now.ToString("hh:mm tt")) + " WHERE No
= " + No;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
SQLCore.Cmd.ExecuteNonQuery();
```

```

SQLCore.cmdText = "Select * from test_db.tb_historydata where Date = " +
DateTime.Now.ToString("dd/MMM/yy") + """;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
MySqlDataReader reader = SQLCore.Cmd.ExecuteReader();
int count = 1;
while (reader.Read())
{
Time = (string)reader["TimeFinish"];
batch = (string)reader["Lot_Batch"];
MFG = (string)reader["MFG"];
Vin = (string)reader["Vin_No"];
form_msg = "Time " + Time + "._" + AddOrdinal(count) + " Unit\n" +
"Batch#" + batch + "_MFG." + MFG + "\n" +
"Vin#" + Vin + "\n\n";
items_message.Add(form_msg);
count++;
}

SQLCore.Conn.Close();
SM_ACTUAL.Content = ACTUAL.Text;
SM_PLAN.Content = PLAN.Text.Trim();
SM_DIFF.Content = Convert.ToInt32(ACTUAL.Text) - Convert.ToInt32(PLAN.Text);
int Diff = Convert.ToInt32(SM_DIFF.Content);
if (Diff>=0)
{
SM_DIFF.Foreground = new SolidColorBrush(Color.FromArgb(255, 102, 255, 51));
MainWindow.P1_ARQ.Input_Diff.Foreground = new
SolidColorBrush(Color.FromArgb(255, 102, 255, 51));
}
else
{

```

```

SM_DIFF.Foreground = new SolidColorBrush(Color.FromArgb(255, 255, 0, 0));
MainWindow.P1_ARQ.Input_Diff.Foreground = new
SolidColorBrush(Color.FromArgb(255, 255, 0, 0));
}

if (StateLineInput == "ARQ")
{
#region paren page1 ARQ
MainWindow.P1_ARQ.Input_Actual.Content = ACTUAL.Text.Trim();
MainWindow.P1_ARQ.Input_Plane.Content = PLAN.Text.Trim();
MainWindow.P1_ARQ.Input_Diff.Content = SM_DIFF.Content;
#endregion
}
else if (StateLineInput == "SubAssy")
{
#region paren page1 SubAssy
MainWindow.P1_SubAssy.Input_Actual.Content = ACTUAL.Text.Trim();
MainWindow.P1_SubAssy.Input_Plane.Content = PLAN.Text.Trim();
MainWindow.P1_SubAssy.Input_Diff.Content = Convert.ToInt32(ACTUAL.Text) -
Convert.ToInt32(PLAN.Text);

#endregion
}

else if (StateLineInput == "SKD")
{
#region paren page1 SKD
MainWindow.P1_SKD.Input_Actual.Content = ACTUAL.Text.Trim();
MainWindow.P1_SKD.Input_Plane.Content = PLAN.Text.Trim();

```

```
MainWindow.P1_SKD.Input_Diff.Content = Convert.ToInt32(ACTUAL.Text) -
Convert.ToInt32(PLAN.Text);
```

```
#endregion
}
```

```
for (int i = items_message.Count - 1; i >= 0; i--)
{
    msg += items_message[i];
}
_lineNotify(start + msg, 0, 0, "");
Inint();
}
```

```
private string AddOrdinal(int num)
{
```

```
    if (num <= 0) return num.ToString();
```

```
    switch (num % 100)
```

```
    {
```

```
        case 11:
```

```
        case 12:
```

```
        case 13:
```

```
        return num + "th";
```

```
    }
```

```
    switch (num % 10)
```

```
    {
```

```
        case 1:
```

```
        return num + "st";
```

```

case 2:
return num + "nd";
case 3:
return num + "rd";
default:
return num + "th";
}
}
}
}

```

4. โค้ดหน้าป้อนข้อมูลการเกิดเบรกดาว

```

using Microsoft.Win32;
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Tulpep.NotificationWindow;
using EExcel = Microsoft.Office.Interop.Excel;

```

```

namespace Datadisplay_Rev4
{

    public partial class Page3InputData : Page
    {
        List<TodoltemP3> items = new List<TodoltemP3>();
        public string StateLineInput ;
        private bool ButtomUpdate = false;
        public int St_ID;

        public Page3InputData()
        {
            InitializeComponent();
            txDe.IsEnabled = false;
            Bt_Edit.IsEnabled = false;
            Bt_Deleat.IsEnabled = false;
            line.token = "7BdkwzG9YGPh9LqqtCV67ZUwn7HkYYKcsElVCN0pbTD";
            //9lQ4ktApiRfhOiYN9zr9Dmql4kj1DHC7FGWTEaUFAKg //RMA

        }
        class TodoltemP3
        {
            public string No { get; set; }
            public string Date { get; set; }
            public string Start { get; set; }
            public string Stop { get; set; }
            public string Line { get; set; }
            public string Detail { get; set; }
            public string Total { get; set; }
            public string Detel { get; set; }
        }
    }
}

```

```

public string Station { get; set; }
public int NoSQL { get; set; }
public string Status { get; set; }
}

```

```

private void CmDT_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    string str;
    ComboBoxItem item = (ComboBoxItem)CmDT.SelectedItem;
    str = item.Content.ToString();

```

```

    txDe.IsEnabled = (str == "ORTHER") ? true : false;
}

```

```

private void CmDT_Selected(object sender, RoutedEventArgs e)
{
    MessageBox.Show("show : " + txDe.Text);
}

```

```

private void Bt_insert_Click(object sender, RoutedEventArgs e)
{
    if (stationP.Text == null)
    {
        MessageBox.Show("กรุณากำหนดข้อมูล Station");
    }
    else if (StartPk.Text == null)

```

```

{
    MessageBox.Show("กรุณากดปุ่มข้อมูล StartBreakdown");
}
else if (TP_Detail == null)
{
    MessageBox.Show("กรุณากดปุ่มข้อมูล TOPIC DETEAL");
}
else
{
    #region MainInsert
    string strTotal = "";
    string strDetail;
    string stateBreakdown = "";
    if (StopPk.Text == null || StopPk.Text == "")
    {
        strTotal = "";
        StopPk.Text = "";
    }
    else
    {
        TimeSpan elapsed =
            DateTime.Parse(StopPk.Text).Subtract(DateTime.Parse(StartPk.Text));
        strTotal = Convert.ToInt32(elapsed.TotalHours).ToString("D2") + "hr:" +
            elapsed.Minutes.ToString("D2") + " min.";
    }

    strDetail = (CmDT.Text == "ORTHER") ? TP_Detail.Text + " : " + "ORTHER : " +
        txDe.Text.Trim() : TP_Detail.Text + " : " + CmDT.Text;

    DateTime _date = DateTime.Now;
    stateBreakdown = (StopPk.Text == "") ? "BREAKDOWN" : "RUNNING";

```

```

SQLCore.Conn.Open();
if (ButtomUpdate == true)
{
    SQLCore.cmdText = SQLCore.cmdText = "UPDATE test_db.tb_historybreakdown SET
    StartBreakdown= " + StartPk.Text.Trim() + ", StopBreakdown= " + StopPk.Text.Trim()
    + ",TotalBreakdown = " + strTotal + ",Detail = " + strDetail + ",Station = " +
    stationP.Text + ",Status = " + stateBreakdown + " WHERE id = " + St_ID;
    SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
    SQLCore.Cmd.ExecuteNonQuery();
    ButtomUpdate = false;

    if (CheckForInternetConnection())
    {
        string msg1 = "\nStation: " + stationP.Text + "\n" + "Status: RUNNING \n" + "Detail: " +
        strDetail + "\n" + "Time: " + StopPk.Text + "\nTotal Breakdown : "+ strTotal;
        lineNotify(msg1);
        string url = ("https://i.ibb.co/pyXs3sZ/Untitled-2-02-04-03.png");
        _lineNotify(" ", 0, 0, url);
    }
    else
    {
        SQLCore.cmdText = "INSERT INTO test_db.tb_historybreakdown
        (Date,StartBreakdown,StopBreakdown,TotalBreakdown,Detail,Line,Station,Status)
        VALUES (?Date,?StB,?StopB,?TtB,?DT,?Line,?St,?Status)";
        SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
        SQLCore.Cmd.Parameters.AddWithValue("@Date", _date.ToString("dd/MMM/yy"));
        SQLCore.Cmd.Parameters.AddWithValue("@StB", StartPk.Text);
        SQLCore.Cmd.Parameters.AddWithValue("@StopB", StopPk.Text);
        SQLCore.Cmd.Parameters.AddWithValue("@TtB", strTotal);
    }
}

```

```

SQLCore.Cmd.Parameters.AddWithValue("@DT", strDetail);
SQLCore.Cmd.Parameters.AddWithValue("@Line", StateLineInput);
SQLCore.Cmd.Parameters.AddWithValue("@St", stationP.Text);
SQLCore.Cmd.Parameters.AddWithValue("?Status", stateBreakdown);
SQLCore.Cmd.ExecuteNonQuery();
if (CheckForInternetConnection())
{
    string msg1 = "\nStation :" + stationP.Text + "\n" + "Status : BREAKDOWN \n" + "Detail : " + strDetail + "\n" + "Time : " + StartPk.Text;
    lineNotify(msg1);
    string url = ("https://i.ibb.co/vxYq3pV/Untitled-2-02-04-04.png");
    _lineNotify(" ", 0, 0,url );
}
}
SQLCore.Conn.Close();
Inint();
ClearAll();
Bt_insert.Content = "Insert";
Bt_insert.IsEnabled = true;
Bt_Edit.IsEnabled = false;
Bt_Deleat.IsEnabled = false;
//MainWindow.P2_ARQ.LoadMap();
#endregion
}
}

public void Inint() //พวกับ MainUserWindow
{
    stationP.Items.Clear();
    if (StateLineInput == "ARQ")
    {

```

```
foreach (LineProduction line in MainWindow.Xml.listLineProduction)
{
    foreach (Block bc in line.listBlock)
    {
        if (bc.Type == "car")
        {
            stationP.Items.Add(bc.Name);
        }
    }
}

if (StateLineInput == "SubAssy")
{
    foreach (LineProduction line in MainWindow.Xml_SubAssy.listLineProduction)
    {
        foreach (Block bc in line.listBlock)
        {
            if (bc.Type == "car")
            {
                stationP.Items.Add(bc.Name);
            }
        }
    }
}

if (StateLineInput == "SKD")
{
    foreach (LineProduction line in MainWindow.Xml_SKD.listLineProduction)
    {
        foreach (Block bc in line.listBlock)
        {
```

```

if (bc.Type == "car")
{
    stationP.Items.Add(bc.Name);
}
}
}
}
}

```

```

for (int i = items.Count - 1; i >= 0; i--)
{
    items.RemoveAt(i);
}

SQLCore.Conn.Open();//เปิดข้อมูลขึ้นมา
SQLCore.cmdText = "Select * from test_db.tb_historybreakdown where Line = " +
    StateLineInput + """;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
MySQLDataReader reader = SQLCore.Cmd.ExecuteReader();
int countRow = 1;
while (reader.Read())
{
    DateTime _date = DateTime.Now;
    items.Add(new TodoItemP3()
    {
        No = countRow.ToString(),
        Date = (string)reader["Date"],
        Start = (string)reader["StartBreakdown"],
        Stop = (string)reader["StopBreakdown"],
        Total = (string)reader["TotalBreakdown"],
        Detail = (string)reader["Detail"],
    });
}

```

```

Line = (string)reader["Line"],
Station = (string)reader["Station"],
NoSQL = (int)reader["id"],
Status = (string)reader["Status"]
});
countRow++;
}
SQLCore.Conn.Close();
DataListDt.ItemsSource = null;
DataListDt.ItemsSource = items;

if (StateLineInput == "ARQ")
{

MainWindow.P2_ARQ.stateMap = "ARQ";
MainWindow.P2_ARQ.LoadMap();
MainWindow.P3_ARQ.stateSQL = "ARQ";
MainWindow.P3_ARQ.Inint();

}
else if (StateLineInput == "SubAssy")
{
MainWindow.P2_SubAssy.stateMap = "SubAssy";
MainWindow.P2_SubAssy.LoadMap();
MainWindow.P3_SubAssy.stateSQL = "SubAssy";
MainWindow.P3_SubAssy.Inint();
}
else if (StateLineInput == "SKD")
{
MainWindow.P3_SKD.stateSQL = "SKD";

```

```

MainWindow.P2_SKD.LoadMap();
MainWindow.P3_SKD.stateSQL = "SKD";
MainWindow.P3_SKD.Inint();

}
else
{

}
// MainWindow.loadstatus();

}

private void Bt_Edit_Click(object sender, RoutedEventArgs e)
{
try
{
string textDT;
string[] str;
textDT = (items[DataListDt.SelectedIndex].Detail);
str = textDT.Split(":").ToArray();
St_ID = (items[DataListDt.SelectedIndex].NoSQL);
stationP.Text = (items[DataListDt.SelectedIndex].Station);
StartPk.Text = (items[DataListDt.SelectedIndex].Start);
StopPk.Text = (items[DataListDt.SelectedIndex].Stop);
TP_Detail.Text = Convert.ToString(str[0]).Trim();

if (str.Length == 3)
{
txDe.Text = Convert.ToString(str[2]).Trim();
CmDT.Text = "";

```

```

    }

    CmDT.Text = Convert.ToString(str[1]).Trim();

    Bt_insert.Content = "Update";
    ButtomUpdate = true;
}
catch (Exception )
{
    MessageBox.Show("ไม่พบข้อมูลให้แก้ไข");
}

Bt_Edit.IsEnabled = false;
Bt_Deleat.IsEnabled = false;
Bt_insert.IsEnabled = true;

}

private void Bt_Deleat_Click(object sender, RoutedEventArgs e)
{
    St_ID = (items[DataListDt.SelectedIndex].NoSQL);
    SQLCore.Conn.Open();
    SQLCore.cmdText = SQLCore.cmdText = "DELETE FROM test_db.tb_historybreakdown
    WHERE id =" + St_ID.ToString();
    SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
    SQLCore.Cmd.ExecuteNonQuery();

    SQLCore.Conn.Close();
    Inint();
    Bt_insert.IsEnabled = true;
    Bt_Edit.IsEnabled = false;
    Bt_Deleat.IsEnabled = false;
    // MainWindow.P2_ARQ.LoadMap();

```

```

}

private void Bt_Cancel_Click(object sender, RoutedEventArgs e)
{
    //ClearAll();

    Bt_insert.IsEnabled = true;
    Bt_Edit.IsEnabled = false;
    Bt_Deleat.IsEnabled = false;

}

private void ClearAll()
{
    Bt_insert.Content = "Insert";
    stationP.Text = "";
    StartPk.Text = "";
    StopPk.Text = "";
    TP_Detail.Text = "";
    Bt_Edit.IsEnabled = true;
    Bt_Deleat.IsEnabled = true;
}

private void DataListDt_SelectionChanged(object sender, SelectionChangedEventArgs
e)
{
    Bt_Edit.IsEnabled = true;
    Bt_Deleat.IsEnabled = true;
    Bt_insert.IsEnabled = false;
}

private void Page_Loaded(object sender, RoutedEventArgs e)
{

```

```

Bt_Edit.IsEnabled = false;
Bt_Deleat.IsEnabled = false;
}

```

```

private void Export_Click(object sender, RoutedEventArgs e)
{
    SaveFileDialog sfd = new SaveFileDialog()
    {
        Title = "เลือกบันทึกไฟล์",
        Filter = "Excel Workbook|*.xls",
        ValidateNames = true
    };

```

```

    if (sfd.ShowDialog() == true)
    {
        EExcel.Application excelApp = new EExcel.Application();
        if (excelApp != null)
        {
            EExcel.Workbook excelWorkbook = excelApp.Workbooks.Add();
            //EExcel.Worksheet excelWorksheet = (EExcel.Worksheet)excelWorkbook.Sheets.Add();
            EExcel.Worksheet excelWorksheet = (EExcel.Worksheet)excelWorkbook.Sheets[1];
            //ws = wb.Worksheets[Sheet];

```

```

            excelWorksheet.Cells[1, 1] = "No";
            excelWorksheet.Cells[1, 2] = "Date";
            excelWorksheet.Cells[1, 3] = "Start";
            excelWorksheet.Cells[1, 4] = "Total";
            excelWorksheet.Cells[1, 5] = "Detail";
            excelWorksheet.Cells[1, 6] = "Line";

```

```

excelWorksheet.Cells[1, 7] = "Station";

SQLCore.Conn.Open();
SQLCore.cmdText = "Select * from test_db.tb_historybreakdown where Line = " +
StateLineInput + """;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
MySQLDataReader reader = SQLCore.Cmd.ExecuteReader();
int countRow = 2;
while (reader.Read())
{
    excelWorksheet.Cells[countRow, 1] = Convert.ToString(countRow - 1);
    excelWorksheet.Cells[countRow, 2] = (string)reader["Date"];
    excelWorksheet.Cells[countRow, 3] = (string)reader["StartBreakdown"];
    excelWorksheet.Cells[countRow, 4] = (string)reader["TotalBreakdown"];
    excelWorksheet.Cells[countRow, 5] = (string)reader["Detail"];
    excelWorksheet.Cells[countRow, 6] = (string)reader["Line"];
    excelWorksheet.Cells[countRow, 7] = (string)reader["Station"];
    //excelApp.ActiveWorkbook.Save();
    countRow++;
}
SQLCore.Conn.Close();
excelApp.ActiveWorkbook.SaveAs(sfd.FileName,
EExcel.XlFileFormat.xlWorkbookNormal);
excelWorkbook.Close();
excelApp.Quit();
}
}
}

public static bool CheckForInternetConnection()
{

```

```

try
{
    using (var client = new WebClient())
    using (client.OpenRead("http://google.com/generate_204"))
    return true;
}
catch
{
    return false;
}
}

private void _lineNotify(string msg, int stickerPackageID, int stickerID, string pictureUrl)
{
    try
    {
        var request = (HttpWebRequest)WebRequest.Create("https://notify-
        api.line.me/api/notify");

        var postData = string.Format("message={0}", msg);
        if (stickerPackageID > 0 && stickerID > 0)
        {
            var stickerPackageId = string.Format("stickerPackageId={0}", stickerPackageID);
            var stickerId = string.Format("stickerId={0}", stickerID);
            postData += "&" + stickerPackageId.ToString() + "&" + stickerId.ToString();
        }
        if (pictureUrl != "")
        {
            var imageThumbnail = string.Format("imageThumbnail={0}", pictureUrl);
            var imageFullsize = string.Format("imageFullsize={0}", pictureUrl);
            postData += "&" + imageThumbnail.ToString() + "&" + imageFullsize.ToString();
        }
    }
    catch { }
}

```

```

    }

    var data = Encoding.UTF8.GetBytes(postData);

    request.Method = "POST";
    request.ContentType = "application/x-www-form-urlencoded";
    request.ContentLength = data.Length;
    request.Headers.Add("Authorization", "Bearer " + line.token);

    using (var stream = request.GetRequestStream()) stream.Write(data, 0, data.Length);
    var response = (HttpWebResponse)request.GetResponse();
    var responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.ToString());
    }
}

private void notifyPicture(string url)
{
    _lineNotify(" ", 0, 0, url);
}

private void notifySticker(int stickerID, int stickerPackageID)
{
    _lineNotify(" ", stickerPackageID, stickerID, "");
}

private void lineNotify(string msg)
{
    _lineNotify(msg, 0, 0, "");
}
}

```

5. โค้ดหน้าป้อนข้อมูลการพักเบรก

```

using MySql.Data.MySqlClient;
using System.Windows;

namespace Datadisplay_Rev4
{
    /// <summary>
    /// Interaction logic for settimebleak.xaml
    /// </summary>
    public partial class settimebleak : Window
    {
        int No;
        public settimebleak()
        {
            InitializeComponent();

            SQLCore.Conn.Open();
            SQLCore.cmdText = "SELECT* From test_db.tb_settimebreak WHERE state = " +
                "first" + "";
            SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
            SQLCore.dr = SQLCore.Cmd.ExecuteReader();
            while (SQLCore.dr.Read())
            {
                No = (int)SQLCore.dr["No"];
                TxtStartMoningBreak.Text = (string)SQLCore.dr["StartMoningBreak"];
                TxtEndMoningBreak.Text = (string)SQLCore.dr["EndMoningBreak"];

                TxtStartLunchBreak.Text = (string)SQLCore.dr["StartLunchBreak"];
                TxtEndLunchBreak.Text = (string)SQLCore.dr["EndLunchBreak"];

                TxtStartAfternoonBreak.Text = (string)SQLCore.dr["StartAfternoonBreak"];
            }
        }
    }
}

```

```
TxtEndAfternoonBreak.Text = (string)SQLCore.dr["EndAfternoonBreak"];
```

```
TxtStartOvertimeBreak.Text = (string)SQLCore.dr["StartOvertimeBreak"];
```

```
TxtEndOvertimeBreak.Text = (string)SQLCore.dr["EndOvertimeBreak"];
```

```
}
```

```
SQLCore.dr.Close();
```

```
SQLCore.Conn.Close();
```

```
}
```

```
private void closeAdd_Click_1(object sender, RoutedEventArgs e)
```

```
{
```

```
    this.Close();
```

```
}
```

```
private void save_Click(object sender, RoutedEventArgs e)
```

```
{
```

```
    SQLCore.Conn.Open();
```

```
    string StartMoningBreak = TxtStartMoningBreak.Text.Trim();
```

```
    string EndMoningBreak = TxtEndMoningBreak.Text.Trim();
```

```
    string StartLunchBreak = TxtStartLunchBreak.Text.Trim();
```

```
    string EndLunchBreak = TxtEndLunchBreak.Text.Trim();
```

```
    string StartAfternoonBreak = TxtStartAfternoonBreak.Text.Trim();
```

```
    string EndAfternoonBreak = TxtEndAfternoonBreak.Text.Trim();
```

```
    string StartOvertimeBreak = TxtStartOvertimeBreak.Text.Trim();
```

```
    string EndOvertimeBreak = TxtEndOvertimeBreak.Text.Trim();
```

```

SQLCore.cmdText = SQLCore.cmdText = "UPDATE test_db.tb_settimebreak SET
StartMoningBreak= " + StartMoningBreak + ", EndMoningBreak= " + EndMoningBreak
+ ",StartLunchBreak = " + StartLunchBreak + ",EndLunchBreak = " + EndLunchBreak
+ ",StartAfternoonBreak = " + StartAfternoonBreak + ",EndAfternoonBreak = " +
EndAfternoonBreak + ",StartOvertimeBreak = " + StartOvertimeBreak +
",EndOvertimeBreak = " + EndOvertimeBreak + " WHERE No = " + No;
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
SQLCore.Cmd.ExecuteNonQuery();
SQLCore.Conn.Close();
MainWindow.StateSettimeBreak = "Settime";
MessageBox.Show("บันทึกข้อมูลสำเร็จ");
this.Close();
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    this.Close();
}
}
}
}

```

6. โค้ดหน้าแสดงผลในสายการผลิต

```

using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Threading;

namespace Datadisplay_Rev4
{
    /// <summary>
    /// Interaction logic for Page1.xaml
    /// </summary>
    public partial class Page1 : Page

```

```

{

    public Label Input_Plane ;
    public Label Input_Actual;
    public Label Input_Diff;
    public Label Input_TakeTime;
    public Label Input_Model;
    public Label Input_Stn;
    public Label Input_Lot;
    public Label Input_MFG;
    public Label Input_Vin;
    public Label Input_Textwork;
    public Label Input_CountCar;
    public DispatcherTimer Timer;
    public int minute=59;
    public Double Int_TakeTime= 59;

    private int time = 60;

    public Page1()
    {
        InitializeComponent();
        //Input_Timer = Timer;
        Input_Plane = TextPlan;
        Input_Actual = TextActual;
        Input_Diff = TextDiff;
        Input_TakeTime = TextTakt;
        Input_Model = TextModel;
        Input_Stn = TextStn;
        Input_Lot = TextLot;
        Input_MFG = TextMFG;
    }
}

```

```

Input_Vin = TextVin;
Input_Textwork = TextWork;
//Input_CountCar = TextCountCar;
grid_ed.Visibility = Visibility.Hidden;
TBCountDown.Visibility = Visibility.Hidden;
Timer = new DispatcherTimer();
Timer.Interval = new TimeSpan(0, 0, 1);
Timer.Tick += Timer_Tick;
Timer.Start();
}
void Timer_Tick(object sender, EventArgs e)
{
    //Sounsucccess.Stop();
    if (minute <= 0 && time <= 0)
    {
        minute = 59;
        //Sounsucccess.Play();
    }
    else if (time == 0)
    {
        minute--;
        time = 60;
    }
    else
    {
        time--;
        TBCountDown.Content = minute.ToString("D2") + ":" + (time % 60).ToString("D2");
        ARC1.EndAngle = Convert.ToDouble(minute / Int_TakeTime) * 360;
    }
}
}

```

```

    }
}

```

7. โค้ดหน้าแสดงผลสถานะสายการผลิต

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Threading;

namespace Datadisplay_Rev4
{
    /// <summary>
    /// Interaction logic for Page2_ARQ.xaml
    /// </summary>
    public partial class Page2_ARQ : Page
    {

        List<string> colorsList = new List<string>();
        List<int> ListRow = new List<int>();
        List<int> ListColumn = new List<int>();
    }
}

```

```
List<int> ListRowEn = new List<int>();
List<int> ListColumnEn = new List<int>();
List<string> SBreakdown = new List<string>();
List<string> DetailBd = new List<string>();
```

```
public string stateMap;
public bool StateImageShow = false;
public string PartNameImage= "MAINTENANCE";
int count = 0;
int i=0;
```

```
private DispatcherTimer Timer;
private DispatcherTimer TimerLine;
public Page2_ARQ()
{
    InitializeComponent();
    Timer = new DispatcherTimer();
    TimerLine = new DispatcherTimer();
    Timer.Interval = new TimeSpan(0, 0, 1);
    TimerLine.Interval = new TimeSpan(0, 0, 2);
    Timer.Tick += TimerAnimation;
    TimerLine.Tick += Timer_Tick;
    TimerLine.Tick += Guideline;
    Timer.Start();
    TimerLine.Start();
    LoadMap();
}
```

```
private void Timer_Tick(object sender, EventArgs e)
```

```

{

//Console.WriteLine("SBreakdown Count: " + SBreakdown.Count.ToString());
if (SBreakdown.Count > 0)
{
    bd.Visibility = Visibility.Hidden;
    ImEngineer.Visibility = Visibility.Hidden;
    PartNameImage = DetailBd[0];
    StateImageShow = true;
}
else
{
    if (count >= ListRow.Count && count >= ListColumn.Count)
    {
        count = 0;
    }
    bd.Visibility = Visibility.Visible;
    ImEngineer.Visibility = Visibility.Visible;
    int row = ListRow[count];
    int column = ListColumn[count];
    int rowEn = ListRowEn[count];
    int columnEn = ListColumnEn[count];

    Grid.SetRow(bd, row);
    Grid.SetColumn(bd, column);
    Grid.SetRow(ImEngineer, rowEn);
    Grid.SetColumn(ImEngineer, columnEn);
    count++;
}
}

```

```

public void LoadMap()
{
    string[] str;
    string strDt;
    colorsList.Clear();
    ListRow.Clear();
    ListColumn.Clear();
    ListRowEn.Clear();
    ListColumnEn.Clear();
    SBreakdown.Clear();
    DetailBd.Clear();
    StateImageShow = false;

    SQLCore.Conn.Open();
    SQLCore.cmdText = "Select * from test_db.tb_historybreakdown where Line= " +
    stateMap+ " And Status = " + "BREAKDOWN"+ "";
    SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
    MySqlDataReader reader = SQLCore.Cmd.ExecuteReader();
    while (reader.Read())
    {
        colorsList.Add((string)reader["Station"]);
        SBreakdown.Add((string)reader["Status"]);
        strDt = (string)reader["Detail"];
        str = strDt.Split(":".ToCharArray());
        DetailBd.Add(str[1].ToString());
    }
    SQLCore.Conn.Close();

    foreach (LineProduction line in MainWindow.Xml.listLineProduction)
    {

```

```

foreach (Block bc in line.listBlock)
{

#region CreatMap
UserControl1 mm = new UserControl1();
mm.CtStation.Content = bc.Name;
mm.Ctname.Content = bc.NameQC;
mm.SetColor("Green");

foreach (string item in colorsList)
{
if (item == bc.Name)
{
mm.SetColor("Red");
break;
}
}
if (bc.Type == "car")
{
mm.SetType("Car", "image/carV2.png");
ListRow.Add(bc.Row);
ListColumn.Add(bc.Column);
ListRowEn.Add(bc.RowEn);
ListColumnEn.Add(bc.ColumnEn);
}
else if (bc.Type == "Arrow")
{
if (bc.direction == "TurnLeftPoint")
{
mm.SetType("Arrow", "Arrow/ArL.png");
}
}
}

```

```

else if (bc.direction == "TurnRightPoint")
{
mm.SetType("Arrow", "Arrow/ArR.png");
}
else if (bc.direction == "StraightPoint")
{
mm.SetType("Arrow", "Arrow/ArS.png");
}
else if (bc.direction == "TurnLeftNoPoint")
{
mm.SetType("Arrow", "Arrow/ArLn.png");
}
else if (bc.direction == "TurnRightNoPoint")
{
mm.SetType("Arrow", "Arrow/ArRn.png");
}
else if (bc.direction == "StraightNoPoint")
{
mm.SetType("Arrow", "Arrow/ArSn.png");
}
else
{
Console.WriteLine("No Coditoon");
}
}
else
{
}

mm.SetRototion(bc.angle);
G.Children.Add(mm); //เอา im เข้าไปใน Grid
Grid.SetRow(mm, bc.Row);

```

```
Grid.SetColumn(mm, bc.Column);
```

```
#endregion
```

```
}
```

```
}
```

```
}
```

```
private void TimerAnimation(object sender, EventArgs e)
```

```
{
```

```
i++;
```

```
if (i > 3)
```

```
{
```

```
i = 1;
```

```
}
```

```
if (StateImageShow)
```

```
{
```

```
imanimation.Visibility = Visibility.Visible;
```

```
}
```

```
else
```

```
{
```

```
imanimation.Visibility = Visibility.Hidden;
```

```
}
```

```
imanimation.Source = new BitmapImage(new
```

```
Uri(@"animation/" + PartNameImage.Trim() + i + ".png", UriKind.Relative));
```

```
// Console.WriteLine(@"animation/" + PartNameImage.Trim() + i + ".png");
```

```
}
```

```
private void Guideline(object sender, EventArgs e)
```

```
{
```

```

if (StateImageShow)
{
    ImGuideline.Visibility = Visibility.Hidden;
}
else
{
    ImGuideline.Visibility = Visibility.Visible;
}

ImGuideline.Source = new BitmapImage(new Uri(@"Guideline"+stateMap+"/"+ count
+ ".png", UriKind.Relative));
//Console.WriteLine("Guideline" + stateMap + "/" + count + ".png");
}
}
}

```

8. โค้ดหน้าแสดงประวัติการเกิดเบรกดาว

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;

```

```

using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Datadisplay_Rev4
{
    /// <summary>
    /// Interaction logic for Page3.xaml
    /// </summary>

    public partial class Page3 : Page
    {

        List<TodoItem> items = new List<TodoItem>();
        static public string stline;
        public int id;

        public string stateSQL;
        public Page3()
        {
            InitializeComponent();
            DataList.ItemsSource = items;
            ///Inint();
        }
        public void Inint()
        {
            for (int i = items.Count - 1; i >= 0; i--)
            {
                items.RemoveAt(i);
            }
            SQLCore.Conn.Open();//เปิดข้อมูลขึ้นมา

```

```

SQLCore.cmdText = "Select * from test_db.tb_historybreakdown where Line= " +
stateSQL + " And Status = " + "BREAKDOWN" + "";
SQLCore.Cmd = new MySqlCommand(SQLCore.cmdText, SQLCore.Conn);
MySqlDataReader reader = SQLCore.Cmd.ExecuteReader();
int countRow = 1;
while (reader.Read())
{
    DateTime _date = DateTime.Now;
    items.Add(new TodoItem()
    {
        No = countRow.ToString(),
        Date = (string)reader["Date"],
        Start = (string)reader["StartBreakdown"],
        Total = (string)reader["TotalBreakdown"],
        Detail = (string)reader["Detail"],
        Line = (string)reader["Line"],
        Station = (string)reader["Station"],

    });
    countRow++;
}
SQLCore.Conn.Close();
DataList.ItemsSource = null;
DataList.ItemsSource = items;

}

public class TodoItem
{

```

```

public string No { get; set; }
public string Date { get; set; }
public string Start { get; set; }
public string Line { get; set; }

public string Detail { get; set; }
public string Total { get; set; }
public string Detel { get; set; }
public string Station { get; set; }
}
}
}

```

9. โค้ดหน้าแสดงเวลาพักเบรก

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Threading;

namespace Datadisplay_Rev4
{

```

```

/// <summary>
/// Interaction logic for PageBeakdown.xaml
/// </summary>

public partial class PageBeakdown : Page
{
    public int secondprogresbar;
    private int second = 60;
    public int minute;
    public Double Calculatesecondprogresbar;
    private DispatcherTimer Timer;
    int RecodeTime;
    public PageBeakdown()
    {
        InitializeComponent();
        Timer = new DispatcherTimer();
        Timer.Interval = new TimeSpan(0, 0, 1);
        Timer.Tick += Timer_Tick;
        Timer.Start();
        RecodeTime = MainWindow.P1_ARQ.minute;
    }
    void Timer_Tick(object sender, EventArgs e)
    {

        if (minute <= 0 && second <= 0)
        {
            if (MainWindow.stateshowframe == true)
            {
                MainWindow.P1_ARQ.minute = RecodeTime;
                //PageDataInputUser.PD1.minute = RecodeTime;
                minute = minute + 1;
            }
        }
    }
}

```

```

MainWindow.StatePosition = false;
MainWindow.stateshowframe = false;
}
}
else if (second == 0)
{
minute--;
second = 60;
}
else
{
second--;
secondprogresbar--;
clock.Content = minute.ToString("D2") + " : " + (second % 60).ToString("D2");
progress.Value = Convert.ToDouble(secondprogresbar /
Calculatesecondprogresbar) * 100;
}
}
}
}

```

ประวัติย่อของผู้ทำปริญญานิพนธ์

ชื่อ - นามสกุล

นายพรประสิทธิ์ ศุภพล

ประวัติการศึกษา

พ.ศ. 2560 - ปัจจุบัน

วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมหุ่นยนต์

และระบบอัตโนมัติ

สถาบันการจัดการปัญญาภิวัฒน์

พ.ศ. 2557 – 2560

มัธยมศึกษาตอนปลาย สายวิทยาศาสตร์และคณิตศาสตร์

โรงเรียนชัยภูมิภักดีชุมพล

ประวัติการฝึกงาน

พ.ศ. 2563 – 2564

RMA Automotive (RMA group)

Department of Manufacturing Engineer

พ.ศ. 2561 – 2563

Gosoft(Thailand) Co.,Ltd.

Department of Research and development

พ.ศ. 2561

24 Shopping Co., Ltd.

Department of Marketing

อีเมล

phornprasit921@gmail.com