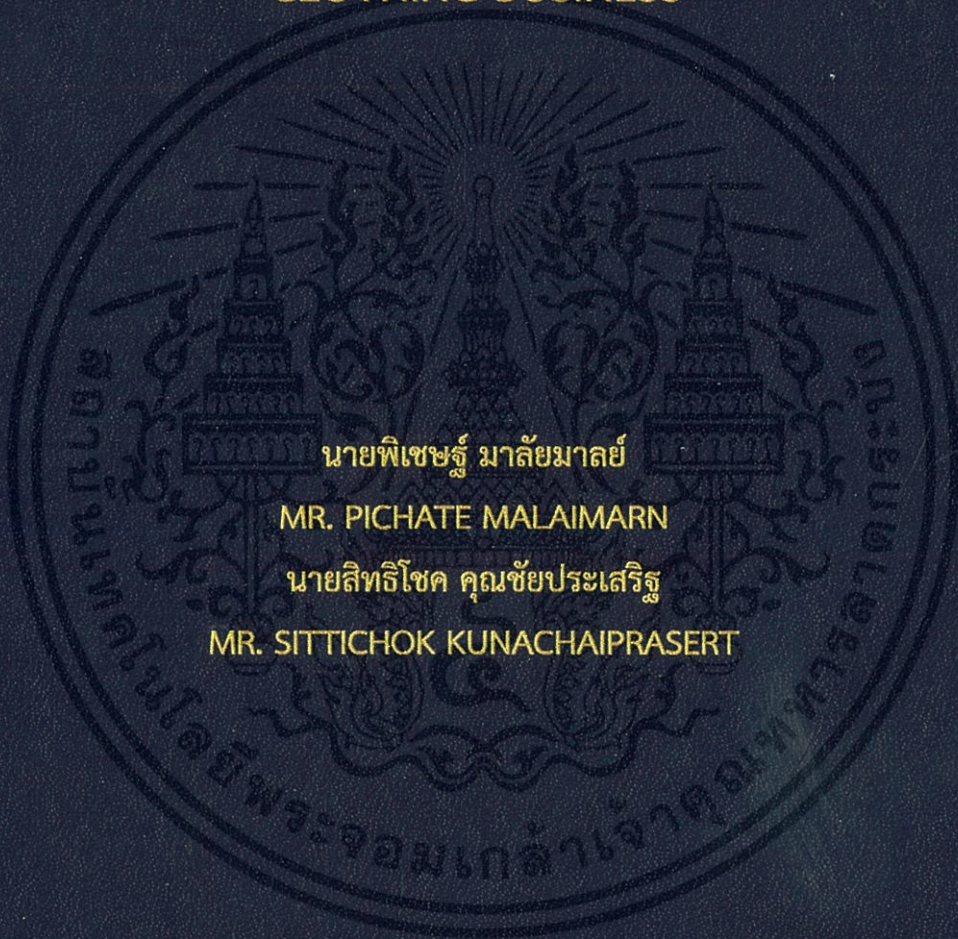


การออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังใน  
ธุรกิจเสื้อผ้าแฟชั่น

THE DESIGN OF A DECISION SUPPORT SYSTEM FOR  
INVENTORY MANAGEMENT: A CASE STUDY OF A  
CLOTHING BUSINESS



นายพิเชษฐ์ มาลัยมัลย์

MR. PICHATE MALAIMARN

นายสิทธิโชค คุณชัยประเสริฐ

MR. SITTICHOK KUNACHAIPRASERT

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

การออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังใน  
ธุรกิจเสื้อผ้าแฟชั่น

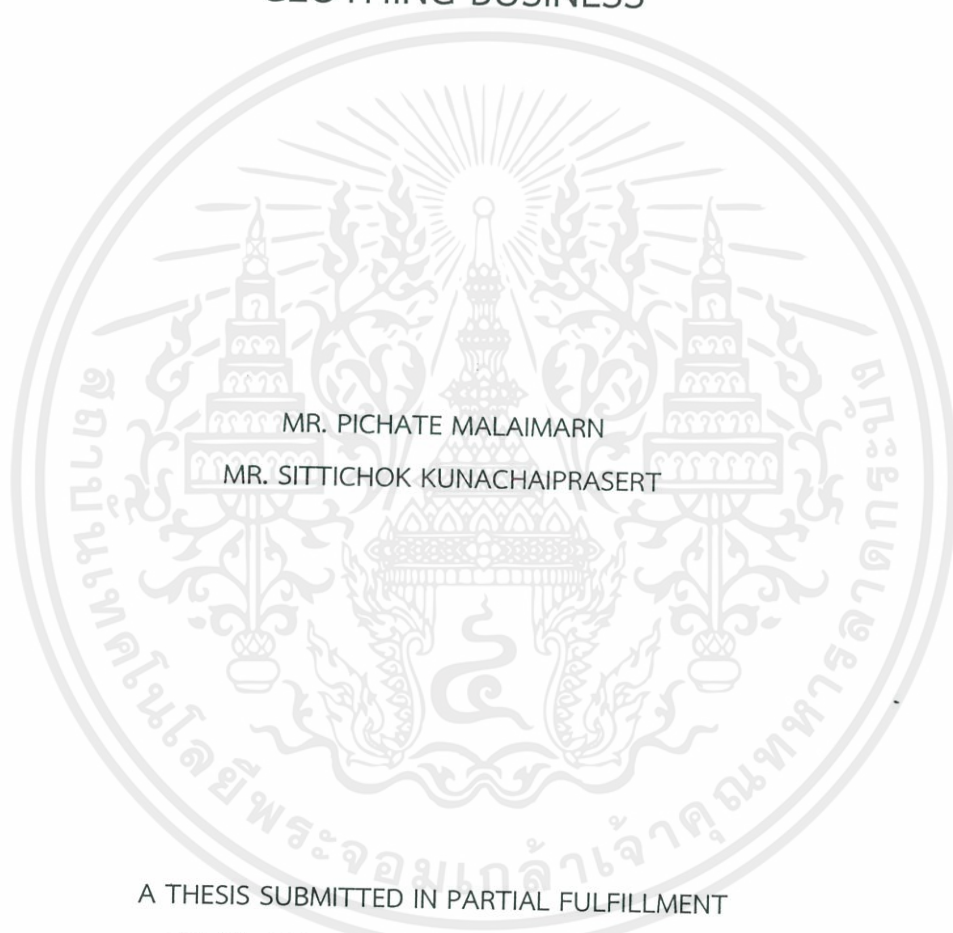
THE DESIGN OF A DECISION SUPPORT SYSTEM FOR  
INVENTORY MANAGEMENT: A CASE STUDY OF A  
CLOTHING BUSINESS

นายพิเชษฐ์ มาลัยมาลย์  
MR. PICHATE MALAIMARN  
นายสิทธิโชค คุณชัยประเสริฐ  
MR. SITTICHOK KUNACHAIPRASERT

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE DESIGN OF A DECISION SUPPORT SYSTEM FOR  
INVENTORY MANAGEMENT: A CASE STUDY OF A  
CLOTHING BUSINESS



MR. PICHATE MALAIMARN  
MR. SITTICHOK KUNACHAIPRASERT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INDUSTRIAL ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
ACADAMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์

การออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังใน  
ธุรกิจเสื้อผ้าแฟชั่น

THE DESIGN OF A DECISION SUPPORT SYSTEM FOR INVENTORY  
MANAGEMENT: A CASE STUDY OF A CLOTHING BUSINESS

นักศึกษา

นายพิเชษฐ์ มาลัยมัลย์ รหัสประจำตัวนักศึกษา 56010855  
นายสิทธิโชค คุณชัยประเสริฐ รหัสประจำตัวนักศึกษา 56011303

หลักสูตร

วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหกรรม

อาจารย์ผู้ควบคุมปริญญาานิพนธ์

(อาจารย์รณน เจียรตระกูล)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น
นักศึกษา	นายพิเชษฐ์ มาลัยมัลย์ นายสิทธิโชค คุณชัยประเสริฐ
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2559
อาจารย์ผู้ควบคุมปริญญานิพนธ์	อาจารย์รณณ เจียรตระกูล

#### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น โดยมีวัตถุประสงค์เพื่อออกแบบและพัฒนาเว็บแอปพลิเคชันการจัดการฐานข้อมูลในการสนับสนุนการตัดสินใจการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น โดยในการสร้างเว็บแอปพลิเคชันผ่านทางโปรแกรม NetBeans IDE 8.1 ประกอบไปด้วย ส่วนของผู้ดูแลระบบ ส่วนของการบันทึกข้อมูล ส่วนของรายละเอียดสินค้า ส่วนของการจัดการสินค้าคงคลังและส่วนของสถิติซื้อขายสินค้า และสามารถนำเว็บแอปพลิเคชันสนับสนุนการตัดสินใจไปใช้ในการบริหารจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่นจากที่ออกแบบไว้ได้อย่างมีประสิทธิภาพ ผลการทดสอบการทำงาน และประสิทธิภาพในระบบการสนับสนุนการตัดสินใจแสดงให้เห็นว่าสถานะสินค้าเพียงพอต่อการขายหรือไม่ ถ้าสินค้าไม่เพียงพอต้องสั่งสินค้าในจำนวนเท่าใด และอัตราการหมุนเวียนของสินค้ามีการเคลื่อนที่เร็ว ปานกลาง หรือช้า ซึ่งเป็นตัวบ่งบอกถึงความต้องการของผู้บริโภคในสินค้าแต่ละแบบว่าขายดีหรือไม่ดี โดยที่ส่งผลถึงระยะเวลาในการขายสินค้า ซึ่งผลที่ได้จะระบุถึงประสิทธิภาพของโปรแกรมว่า มีสินค้าคงคลังลดลงเข้าใกล้จำนวนความต้องการสินค้าของผู้บริโภค ทำให้ลดการเกิดสินค้าที่มีมากเกินไปในคลัง และสินค้าคงคลังไม่ต่ำกว่าจำนวนความต้องการสินค้าของผู้บริโภค ทำให้ไม่เกิดปัญหาการขาดสินค้า ส่งผลให้ประสิทธิภาพในการจัดการโดยรวมเพิ่มมากขึ้น

<b>Thesis Title</b>	The Design of a Decision Support System for Inventory Management: A Case Study of a Clothing Business
<b>Student</b>	Mr. Pichate Malaimarn Mr. Sittichok Kunachaiprasert
<b>Degree</b>	Bachelor of Engineering in Industrial Engineering King Mongkut's Institute of Technology Ladkrabang
<b>Academic year</b>	2016
<b>Thesis Advisor</b>	Mr. Ranon Jientrakul

### ABSTRACT

This thesis presents the design of a decision support system for inventory management. The purpose of this project is to design and develop a web application for a decision support database management for inventory management by using program NetBeans IDE 8.1. This software includes system administration section, information receiving section, product detail section and decision support system section. Our software can efficiently be used in clothing business. The results show that that the system can accurately predict the demand, if it is not enough how many products we must order in addition. Moreover, inventory turnover have shown fast, medium, slow, and dead stock in that period. It also provides amount of product sold and suggest which products are popular and which should be discontinued. Additionally, using our software results in the decrease of inventory level and eventually improves overall performance in management.

## กิตติกรรมประกาศ

ปริญญาานิพนธ์เรื่องการออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่นสามารถสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณบุคคลทุกท่านที่มีส่วนเกี่ยวข้องส่งผลให้ปริญญาานิพนธ์นี้เสร็จสมบูรณ์ โดยได้รับคำปรึกษาจาก อาจารย์ธรรณ เจียรตระกูล ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญาานิพนธ์ ดร.เขาวลิต หามนตรี และ ดร.ภาสุ พูนภักดี ซึ่งเป็นอาจารย์ที่ให้การสนับสนุนในการดำเนินการทำปริญญาานิพนธ์ในลำดับถัดมา คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงสำหรับ รวมทั้งความรู้ คำแนะนำ ความช่วยเหลือและความเอาใจใส่ในทุกๆด้านของปริญญาานิพนธ์ตลอดช่วงเวลาที่ผ่านมา และสุดท้ายนี้ขอกราบขอบพระคุณท่านอาจารย์ภายในภาควิชาวิศวกรรมอุตสาหกรรม คุณพ่อ คุณแม่ ครอบครัวอันเป็นที่รัก รวมถึงเพื่อนๆและพี่ๆ สำหรับการให้การสนับสนุน ทั้งด้านความช่วยเหลือ และกำลังใจ การให้คำปรึกษา ทำให้ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

นายพิเชษฐ์ มาลัยมาลัย  
นายสิทธิโชค คุณชัยประเสริฐ

## สารบัญ

	หน้า	
บทคัดย่อภาษาไทย.....	ก	
บทคัดย่อภาษาอังกฤษ.....	ข	
กิตติกรรมประกาศ.....	ค	
สารบัญ.....	ง	
สารบัญตาราง.....	ฉ	
สารบัญรูป.....	ญ	
<b>บทที่ 1</b>	<b>บทนำ</b>	
1.1	ความเป็นมาและความสำคัญของปัญหา.....	1
1.2	วัตถุประสงค์.....	2
1.3	ขอบเขตของปริญญานิพนธ์.....	2
1.4	ประโยชน์ที่คาดว่าจะได้รับ.....	2
<b>บทที่ 2</b>	<b>ทฤษฎีที่เกี่ยวข้อง</b>	
2.1	เว็บแอปพลิเคชัน.....	3
2.1.1	ประโยชน์ของเว็บแอปพลิเคชัน.....	4
2.1.2	Java Server Faces.....	4
2.1.3	ขั้นตอนการทำงานของ Model-View-Controller.....	4
2.1.4	ขั้นตอนการทำงานของ Model-View-Controller.....	5
2.2	ฐานข้อมูล.....	5
2.2.1	ฐานข้อมูลและคลังข้อมูล.....	5
2.2.1.1	โครงสร้างข้อมูล.....	6
2.2.1.2	ปัญหาเกี่ยวกับแฟ้มข้อมูล.....	6
2.2.1.3	แนวทางในการใช้ฐานข้อมูลในการบริหารจัดการข้อมูล.....	7

## สารบัญ (ต่อ)

	หน้า
2.2.2 Database Management System.....	8
2.2.2.1 หน้าที่ของ Database Management System.....	8
2.2.2.2 การจัดการฐานข้อมูล.....	9
2.2.2.3 ส่วนประกอบที่สำคัญในระบบการจัดการข้อมูล.....	9
2.2.2.4 องค์ประกอบของระบบฐานข้อมูล.....	10
2.2.2.5 อุปสรรคในการพัฒนาระบบฐานข้อมูล.....	10
2.2.3 การออกแบบฐานข้อมูล.....	11
2.2.3.1 รูปแบบของฐานข้อมูล.....	11
2.2.3.2 ทฤษฎีวงจรการพัฒนาระบบฐานข้อมูล.....	13
2.2.4 เครื่องมือที่ใช้ในการวิเคราะห์ระบบ.....	15
2.2.4.1 แผนผังการไหลของข้อมูล Flow Chart.....	15
2.2.4.2 อีอาร์โมเดล.....	17
2.2.4.3 ความสัมพันธ์ในฐานข้อมูลเชิงสัมพันธ์.....	18
2.2.4.4 คีย์.....	20
2.2.5 คลังข้อมูล.....	20
2.2.5.1 ลักษณะของคลังข้อมูล.....	21
2.2.5.2 การวิเคราะห์ข้อมูลหลายมิติและการค้นหาความรู้ในคลังข้อมูล.....	22
2.2.5.3 ดาต้าไมนิ่ง.....	22
2.2.5.4 ความแตกต่างระหว่างคลังข้อมูลกับฐานข้อมูลปฏิบัติการ.....	22
2.2.6 การศึกษาการคำนวณพื้นฐานอ้างอิงจากสิ่งที่ผู้ประกอบการต้องการ.....	23
2.2.6.1 การบริหารจัดการสินค้าคงคลัง.....	23
2.2.6.2 การจำแนกหมวดสินค้า.....	23
2.2.6.3 การควบคุมวัสดุคงคลัง.....	24
2.2.6.4 การคงคลังสินค้าสำเร็จรูป.....	24
2.2.6.5 การตัดสินใจขั้นพื้นฐานวัสดุคงคลัง.....	24

## สารบัญ (ต่อ)

	หน้า
2.2.6.6 ข้อควรพิจารณาในการควบคุมวัสดุคงคลัง.....	25
2.2.6.7 ระบบช่วงเวลาการสั่งซื้อคงที่.....	27
2.2.6.8 จุดสั่งซื้อกรณีความต้องการผันแปร.....	27
2.2.6.9 ปริมาณสั่งซื้อกรณีต้องการผันแปร.....	28
2.2.6.10 อัตราการหมุนเวียนของสินค้าคงคลัง.....	29
2.3 NetBeans IDE.....	29
2.3.1 ส่วนประกอบของ NetBeans.....	30
2.3.2 ภาษา JAVA.....	30
2.3.3 MySQL.....	30
<b>บทที่ 3 การออกแบบและวิธีการดำเนินงาน</b>	
3.1 ศึกษารูปแบบธุรกิจ และเก็บข้อมูลปัญหาจากที่เกิดขึ้นจากผู้ประกอบการ.....	32
3.1.1 ข้อมูลและรูปแบบของธุรกิจเสื้อผ้าแฟชั่น.....	32
3.1.2 รูปแบบการบริหารการภายใน.....	33
3.1.3 การรับทราบปัญหาและความต้องการจากผู้ประกอบการ.....	35
3.1.4 การวิเคราะห์จุดบกพร่องการจัดการภายในองค์กร.....	35
3.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับวิทยานิพนธ์ รูปแบบและเครื่องมือที่ใช้ในการแก้ไขปัญหา.....	35
3.2.1 ทฤษฎีที่เกี่ยวข้องกับวิทยานิพนธ์.....	35
3.2.1.1 ระบบช่วงเวลาการสั่งซื้อคงที่.....	35
3.2.1.2 ปริมาณสั่งซื้อกรณีต้องการผันแปร.....	36
3.2.1.3 อัตราการหมุนเวียนของสินค้าคงคลัง.....	36
3.2.2 รูปแบบของการพัฒนาระบบการสนับสนุนการตัดสินใจ.....	36
3.2.2.1 เว็บแอปพลิเคชัน.....	36
3.2.2.2 เครื่องมือที่ใช้ในการสร้างเว็บแอปพลิเคชัน.....	37

## สารบัญ (ต่อ)

หน้า

3.3	จัดการโครงสร้างฐานข้อมูลด้วยรูปแบบ ER Diagram และจัดเรียงข้อมูลผ่าน Microsoft Excel	37
3.3.1	สร้าง ER Diagram และ Flow Chart	38
3.3.2	จัดเรียงข้อมูลผ่าน Microsoft Excel	45
3.4	วางแผนการพัฒนาระบบผ่านรูปแบบของเว็บแอปพลิเคชันโดย NetBeans IDE	45
3.4.1	เก็บความต้องการรูปแบบของหน้าจอแสดงผล	45
3.4.2	ออกแบบหน้าจอ Mock Up ของหน้าตาเว็บแอปพลิเคชัน	46
3.4.3	การกำหนดรหัสของการจัดการสินค้าภายในคลัง	51
3.4.3.1	กำหนดรหัสสินค้าเพื่อแยกประเภทของสินค้าแต่ละโมเดล	51
3.4.3.2	กำหนดรหัสของออเดอร์สินค้า	52
3.4.4	เขียนโปรแกรมทำให้ฟังก์ชันการทำงานภายใน	52
3.5	ทดสอบระบบการทำงานของเว็บแอปพลิเคชัน และสรุปผลการทำงานของระบบ	57
3.5.1	ทดสอบระบบการทำงานของเว็บแอปพลิเคชัน	57
3.5.2	ทดสอบประสิทธิภาพการสนับสนุนการตัดสินใจจากการทำงาน	58

## บทที่ 4 ผลการดำเนินงาน

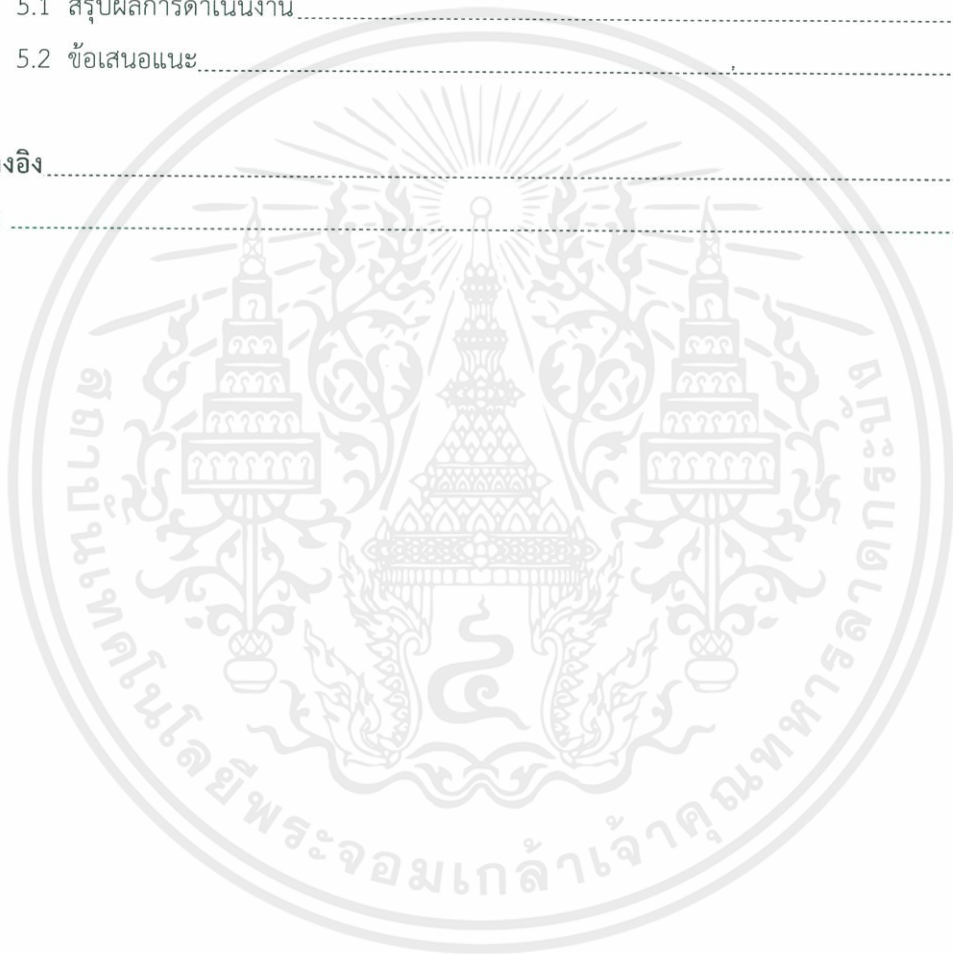
4.1	ศึกษาทฤษฎีที่เกี่ยวข้องกับปริญาณินท์	59
4.2	สร้างระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น	59
4.2.1	ผลการออกแบบและพัฒนาเว็บแอปพลิเคชัน	59
4.2.2	การสร้างระบบการจัดการสินค้าคงคลัง	62
4.2.3	ส่วนการบันทึกและแสดงผลรายละเอียดสินค้า	65
4.2.4	การสร้างส่วนสนับสนุนการตัดสินใจ	66
4.2.5	การวิเคราะห์ข้อมูลจากการขายสินค้า	67
4.2.6	ทดสอบการใช้งานระบบสนับสนุนการตัดสินใจ	68
4.2.7	การเปรียบเทียบประสิทธิภาพในการทำงาน	71

ช

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลการดำเนินงาน	
5.1 สรุปผลการดำเนินงาน.....	96
5.2 ข้อเสนอแนะ.....	97
เอกสารอ้างอิง.....	98
ภาคผนวก.....	ผ1



## สารบัญตาราง

	หน้า
ตารางที่ 2.1 ความหมายของสัญลักษณ์ Flow Chart.....	16
ตารางที่ 2.2 คุณลักษณะของตารางความสัมพันธ์.....	18
ตารางที่ 3.1 Admin.....	42
ตารางที่ 3.2 Model.....	42
ตารางที่ 3.3 Product.....	43
ตารางที่ 3.4 Supplier.....	43
ตารางที่ 3.5 Order.....	44
ตารางที่ 3.6 Order Detail.....	45
ตารางที่ 3.7 Customer.....	45
ตารางที่ 3.8 ตัวอย่างข้อมูลการขายสินค้าเดือนสิงหาคม.....	46
ตารางที่ 3.9 รูปแบบการแสดงผลที่ผู้ประกอบการต้องการเพื่อความง่ายต่อความเข้าใจ.....	46
ตารางที่ 3.10 รายละเอียดรหัสสินค้า.....	52
ตารางที่ 4.1 สินค้ารุ่น 69 กรัม L.....	73
ตารางที่ 4.2 สินค้ารุ่น 69 กรัม M.....	76
ตารางที่ 4.3 สินค้ารุ่น 69 กรัม S.....	79
ตารางที่ 4.4 สินค้ารุ่น 69 กรัม XL.....	82
ตารางที่ 4.5 สินค้ารุ่น 7th Street ต่ำ L.....	85
ตารางที่ 4.6 สินค้ารุ่น 7th Street ต่ำ M.....	88
ตารางที่ 4.7 สินค้ารุ่น 7th Street ต่ำ S.....	91
ตารางที่ 4.8 สินค้ารุ่น 7th Street ต่ำ XL.....	94

## สารบัญรูป

	หน้า
รูปที่ 2.1 ขั้นตอนการทำงานของ Model-View-Controller.....	5
รูปที่ 2.2 ส่วนประกอบของระบบฐานข้อมูลแสดงส่วนประกอบของระบบจัดการฐานข้อมูล.....	10
รูปที่ 2.3 แบบจำลองฐานข้อมูลลำดับชั้น.....	11
รูปที่ 2.4 แบบจำลองฐานข้อมูลเครือข่าย.....	12
รูปที่ 2.5 แบบจำลองฐานข้อมูลเชิงสัมพันธ์.....	13
รูปที่ 2.6 สัญลักษณ์ที่ใช้ในการทำแผนผัง.....	19
รูปที่ 3.1 แผนผังธุรกิจโดยรวม.....	33
รูปที่ 3.2 แผนผังธุรกิจโดยละเอียด.....	34
รูปที่ 3.3 ตัวโปรแกรม NetBeans.....	37
รูปที่ 3.4 ผังเชื่อมโยงความสัมพันธ์ของข้อมูล.....	38
รูปที่ 3.5 แผนภาพการไหลของข้อมูล Flow Chart การเข้าใช้ระบบ.....	39
รูปที่ 3.6 แผนภาพการไหลของข้อมูล Flow Chart ผู้ดูแลระบบ.....	39
รูปที่ 3.7 แผนภาพการไหลของข้อมูล Flow Chart การใช้งานในส่วนของการบันทึกข้อมูล.....	40
รูปที่ 3.8 แผนภาพการไหลของข้อมูล Flow Chart การใช้งานในส่วนของการสนับสนุนการตัดสินใจ.....	40
รูปที่ 3.9 ตัวอย่างข้อมูลภายใน Model.....	41
รูปที่ 3.10 ข้อมูลภายใน Product.....	42
รูปที่ 3.11 ข้อมูลภายใน Supplier.....	43
รูปที่ 3.12 ข้อมูลภายใน Order.....	43
รูปที่ 3.13 ข้อมูลภายใน Order Detail.....	44
รูปที่ 3.14 ข้อมูลภายใน Customer.....	44
รูปที่ 3.15 ส่วนประกอบของโปรเจกต์ในส่วนต่างๆ.....	46
รูปที่ 3.16 ส่วนของสถานะคลังสินค้า Inventory List.....	46
รูปที่ 3.17 ส่วนของรายละเอียดยอดขายสั่งซื้อของสินค้า.....	47
รูปที่ 3.18 ส่วนของรายละเอียดยอดขายสินค้า.....	47

ญ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.19 ส่วนของการบันทึกยอดขายสินค้าผ่านทางหน้าเว็บแอปพลิเคชัน.....	48
รูปที่ 3.20 ส่วนของการบันทึกยอดรับสินค้าผ่านทางหน้าเว็บแอปพลิเคชัน.....	48
รูปที่ 3.21 ส่วนของ Report ส่วนของการสนับสนุนการตัดสินใจ การจัดการคลังสินค้าของผู้ประกอบการ.....	49
รูปที่ 3.22 ส่วนอัตราการผลิตสินค้า.....	49
รูปที่ 3.23 ส่วนของการบันทึกข้อมูลสินค้าเข้าสู่ฐานข้อมูล.....	50
รูปที่ 3.24 รายละเอียดสินค้าที่มีอยู่ในฐานข้อมูลในปัจจุบัน.....	50
รูปที่ 3.25 ส่วนประกอบของ View ที่จะแสดงผลต่อผู้ใช้งาน.....	53
รูปที่ 3.26 ส่วนประกอบของ Controller.....	54
รูปที่ 3.27 ส่วนประกอบภายใน Façade.....	55
รูปที่ 3.28 ส่วนประกอบภายใน Service.....	56
รูปที่ 3.29 ส่วนประกอบภายใน Database.....	57
รูปที่ 4.1 หน้าจอแสดงผลในส่วนของการระบบเพื่อเริ่มต้นการใช้งาน.....	60
รูปที่ 4.2 หน้าจอโดยรวมของเว็บแอปพลิเคชัน.....	60
รูปที่ 4.3 ข้อมูลของผู้ดูแลระบบที่ทำการเข้าสู่ระบบมา.....	61
รูปที่ 4.4 ส่วนของการค้นหาข้อมูลที่ต้องการ.....	61
รูปที่ 4.5 การบันทึกการรับสินค้าเข้าสู่คลัง.....	62
รูปที่ 4.6 รายงานการรับสินค้าหลังจากการบันทึกข้อมูลในส่วนของการบันทึกรับสินค้า.....	63
รูปที่ 4.7 ส่วนการบันทึกข้อมูลการจัดจำหน่ายสินค้า และรายละเอียดที่ต้องกรอกเมื่อจำหน่ายสินค้า.....	64
รูปที่ 4.8 รายงานสถิติยอดขายสินค้า.....	64
รูปที่ 4.9 จำนวนสินค้าภายในคลังสินค้า.....	65
รูปที่ 4.10 ส่วนของการบันทึกข้อมูลสินค้าเข้าสู่ฐานข้อมูล.....	65
รูปที่ 4.11 รายละเอียดสินค้าที่มีอยู่ในฐานข้อมูลในปัจจุบัน.....	66
รูปที่ 4.12 ส่วนข้อมูลที่เป็นประโยชน์ในการตัดสินใจซื้อสินค้าของผู้ประกอบการ.....	67
รูปที่ 4.13 การวิเคราะห์สินค้าจากข้อมูลการขายสินค้า.....	68
รูปที่ 4.14 สินค้าคงคลัง.....	69

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.15 รายงานสถิติยอดการจัดจำหน่ายย้อนหลัง.....	69
รูปที่ 4.16 รายงานสถิติการรับสินค้าย้อนหลัง.....	70
รูปที่ 4.17 ตารางแสดงอัตราการผลิตเวียนสินค้า.....	70
รูปที่ 4.18 ตารางส่วนของการสนับสนุนการตัดสินใจ.....	71
รูปที่ 4.19 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 69 กรัม L.....	74
รูปที่ 4.20 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 69 กรัม M.....	77
รูปที่ 4.21 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 69 กรัม S.....	80
รูปที่ 4.22 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 69 กรัม XL.....	83
รูปที่ 4.23 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 7th Street ดำ L.....	86
รูปที่ 4.24 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 7th Street ดำ M.....	89
รูปที่ 4.25 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 7th Street ดำ S.....	92
รูปที่ 4.26 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 7th Street ดำ XL.....	95

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันความหลากหลายของช่องทางการทำธุรกิจมีมากขึ้นเมื่อเปรียบเทียบกับอดีต สังเกตได้จาก การพัฒนารูปแบบของธุรกิจ โดยเฉพาะอย่างยิ่งในปัจจุบันที่เครือข่ายอินเทอร์เน็ตมีความสำคัญมากในการใช้ ชีวิตประจำวันของมนุษย์ ระบบธุรกิจซื้อขายไปในรูปแบบออนไลน์ทั่วไปได้มีการพัฒนามาอยู่ในรูปแบบของ การซื้อขายออนไลน์ซึ่งระบบอินเทอร์เน็ตเข้ามามีบทบาทเป็นอย่างมาก และเมื่อตลาดการของซื้อขายออนไลน์ เป็นที่แพร่หลาย และได้รับการยอมรับในวงกว้าง อัตราของการแข่งขันทางธุรกิจจึงเพิ่มมากขึ้นในสินค้า ประเภทเดียวกัน เพราะการซื้อขายแบบออนไลน์นั้นมีความสะดวกต่อการเข้าถึงและใช้ต้นทุนในการดำเนิน ธุรกิจที่ต่ำกว่ารูปแบบการซื้อขายแบบเก่า สังเกตได้จากในประเภทธุรกิจสินค้าเดียวกันจะมีคู่แข่งทางการตลาด เป็นจำนวนมาก ดังนั้นการมีสินค้า และระบบการจัดการธุรกิจที่มีประสิทธิภาพจึงมีความสำคัญเพื่อให้ธุรกิจ ก้าวไปสู่ความเป็นผู้นำในธุรกิจสินค้าประเภทนั้นๆ

ธุรกิจเสื้อผ้าแฟชั่นในปัจจุบันเป็นธุรกิจที่ได้รับความนิยมอย่างมากจากผู้ประกอบการ เนื่องจาก กำลังซื้อของกลุ่มลูกค้าวัยรุ่นมีสูงและเสื้อผ้าถือเป็นปัจจัยสี่ในการดำเนินชีวิต เมื่อมีความต้องการซื้อเพิ่ม มากขึ้นส่งผลให้ความต้องการขายมีการเติบโตแปรผันตามปริมาณของความต้องการซื้อเช่นกัน อย่างไรก็ตามแฟชั่น ในปัจจุบันมีการเปลี่ยนแปลงไปอย่างรวดเร็วทำให้ความต้องการของลูกค้ามีการเปลี่ยนแปลงไปตามยุคสมัย ดังนั้นระบบการจัดการสินค้าภายในคลังจึงมีความสำคัญอย่างมากสำหรับผู้ประกอบการในการบริหารจัดการ ธุรกิจ เพื่อลดการจุกเก็บสินค้าที่อาจทำให้มูลค่าลดลง การวางแผนการจัดเก็บสินค้าที่ทำให้เพียงพอต่อ ความต้องการของลูกค้า

รูปแบบการทำธุรกิจเสื้อผ้าแฟชั่นในปัจจุบันมีการซื้อขายสินค้าผ่านทางหลากหลายช่องทาง และการ หมุนเวียนของสินค้าประเภทนี้มีการหมุนเวียนที่รวดเร็ว บางสินค้ามีผลตอบรับที่ดี บางสินค้ามีผลตอบรับที่ไม่ ดี ทำให้การจัดการจัดเก็บสินค้าคงคลังมีความซับซ้อนต่อการคาดคะเนการจัดเก็บสินค้าเพื่อรองรับปริมาณ ความต้องการของลูกค้า ดังนั้น ทางคณะผู้จัดทำได้เล็งเห็นว่ารูปแบบการดำเนินการนั้นสามารถจะพัฒนาให้มี ความสะดวกต่อผู้ประกอบการได้ดีมากยิ่งขึ้น โดยการสร้างระบบสนับสนุนการตัดสินใจการจัดการสินค้าคง คลัง ภายใต้แนวคิดที่รวมข้อมูลเข้าสู่ศูนย์กลาง ผ่านระบบอินเทอร์เน็ตในรูปแบบของเว็บแอปพลิเคชันเพื่อ สร้างระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่นของผู้ประกอบการ

## 1.2 วัตถุประสงค์

1. เพื่อออกแบบและพัฒนาเว็บแอปพลิเคชันการจัดการฐานข้อมูลในการสนับสนุนการตัดสินใจการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น
2. สามารถนำเว็บแอปพลิเคชันสนับสนุนการตัดสินใจไปใช้ในการบริหารจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่นได้อย่างมีประสิทธิภาพ

## 1.3 ขอบเขตของปริญญาบัตร

1. ใช้ข้อมูลการขายจากกรณีศึกษาในช่วงเดือนสิงหาคม 2559
2. ใช้ข้อมูลการขายของเสื้อผ้าแฟชั่นจำนวน 8 รหัสสินค้าเท่านั้น
3. ใช้ข้อมูลการขายจากตัวแทนจำหน่ายในเขตกรุงเทพฯ และปริมณฑลเท่านั้น

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถใช้ระบบสนับสนุนการตัดสินใจคำสั่งซื้อสำหรับการบริหารในภาคธุรกิจเสื้อผ้าแฟชั่นเพื่อความสะดวก และลดข้อผิดพลาดที่อาจเกิดขึ้นจากขั้นตอนคำสั่งซื้อ
2. สามารถประยุกต์ใช้สำหรับภาคธุรกิจอื่น และการพัฒนาโปรแกรมในอนาคต

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

ปฏิญานิพนธ์ฉบับนี้ เป็นการสร้างโปรแกรมเว็บแอปพลิเคชันเพื่อสนับสนุนการตัดสินใจการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น เป็นการศึกษาซึ่งเกี่ยวข้องกับการเขียนโปรแกรมการจัดการฐานข้อมูล และการสร้างรายงานที่มีการผันแปรของสินค้าตลอดเวลาผ่านเว็บแอปพลิเคชัน เพื่อนำไปสู่การจัดการสินค้าคงคลังที่มีประสิทธิภาพ ผลจากการใช้เว็บแอปพลิเคชันดังกล่าวส่งผลให้การจัดการสินค้าคงคลังมีความเหมาะสมกับความต้องการของผู้บริโภคมากขึ้น และสามารถลดปัญหาการเกิดสินค้าค้างสต็อกได้ การทำปฏิญานิพนธ์ฉบับนี้ได้ทำการศึกษาทฤษฎีที่เกี่ยวข้องดังกล่าวในหัวข้อต่อไปนี้

#### 2.1 เว็บแอปพลิเคชัน [1]

เว็บแอปพลิเคชัน คือ การพัฒนาระบบงานบนเว็บ ข้อมูลต่างๆในระบบมีการไหลเวียนในแบบออนไลน์ ทั้งแบบภายในวง LAN และออกไปยังเครือข่ายอินเทอร์เน็ต ทำให้เหมาะสมสำหรับงานที่ต้องการข้อมูลแบบ Real Time ระบบมีประสิทธิภาพ และใช้งานสะดวก ซึ่งระบบสามารถเชื่อมโยงกับผู้ใช้บริการแบบ Real Time

ส่วนประกอบของเว็บแอปพลิเคชัน ในปัจจุบันประกอบไปด้วยการทำงานของเทคโนโลยีต่างๆ มากมาย ไม่ว่าจะเป็นตัวโปรแกรมเว็บแอปพลิเคชัน เว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ซอฟต์แวร์ ฐานข้อมูล เว็บเบราว์เซอร์ และอื่นๆ ซึ่งแต่ละส่วนก็จะมีหน้าที่และการทำงานที่แตกต่างกันออกไป ซึ่งสามารถแยกส่วนประกอบของการทำงานออกเป็นสองส่วนหลักๆ คือ เทคโนโลยีฝั่งผู้ใช้งาน และ เทคโนโลยีฝั่งเซิร์ฟเวอร์ โดยภาษาจาวา เป็นโปรแกรมภาษาที่ถูกพัฒนาเพื่อรองรับการออกแบบซอฟต์แวร์ที่มีการเชื่อมโยงอินเทอร์เน็ต อีกทั้งยังเป็นโปรแกรมที่สนับสนุนแนวคิดของการเขียนโปรแกรมเชิงวัตถุ โดยมีความสามารถเฉพาะตัวต่างจากโปรแกรมภาษาชั้นสูงอื่นๆ เช่น C หรือ C++ ความสามารถดังกล่าวทำให้ สามารถนำไปประยุกต์ใช้งานในเรื่องของการทำงานข้ามระบบปฏิบัติการได้โดยไม่ต้องมีการประมวลใหม่ และโปรแกรมระบบงานจะมีส่วนประกอบที่ทำหน้าที่นำเสนอข้อมูล และรับข้อมูลจากผู้ใช้ซึ่งเรียกว่าเป็น User Interface (UI) ส่วนประกอบนี้มีผลกระทบต่อผู้ใช้งาน คือ การสร้างส่วน UI ให้มีมาตรฐานเป็นรูปแบบของเว็บ (HTML) ซึ่งสามารถเรียกใช้งานด้วยตัวโปรแกรมท่องเว็บทั่วไป

### 2.1.1 ประโยชน์ของเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันเป็นการปรับปรุงเปลี่ยนแปลงรูปแบบจากวินโดวแอปพลิเคชัน โดยใช้เทคนิคที่สามารถรองรับเทคโนโลยีของเว็บได้ คุณสมบัติสำคัญเทคโนโลยีของเว็บดังนี้

1. ใช้ได้ทุกสถานที่ ทุกเวลา ที่สามารถเชื่อมต่ออินเทอร์เน็ต (ปัจจุบันสามารถใช้อินเทอร์เน็ตผ่านระบบ 3G หรือ 4G หรือ WIFI ได้) เพราะ สามารถเข้าใช้งานได้โดยไม่ต้องติดตั้งโปรแกรม
2. ใช้กับอุปกรณ์ไอทีได้หลากหลาย เช่น Computer PC, Notebook, Tablet หรือ มือถือ
3. การใช้งานทำได้สะดวกเพียงมีเว็บเบราว์เซอร์ และข้อมูลจัดเก็บที่เดียว ไม่ทำให้เกิดความซ้ำซ้อน
4. เชื่อมต่อกับเว็บแอปพลิเคชัน หรือบริการออนไลน์อื่นๆได้ง่าย

### 2.1.2 Java Server Faces (JSF)

JSF คือ MVC Web Framework เป็นระบบวิธีคิดที่ออกแบบมาเพื่อแก้ปัญหาที่ต้องการแก้ไข และเป็นระบบความคิดที่ต้องการแยกการทำงานของโปรแกรมส่วนของ Model ออกจากส่วนแสดงผล (View) และมีคนกลางคอยควบคุมความสัมพันธ์ระหว่าง Model และ View คือ Controller อีกทั้งยังช่วยลดความยุ่งยากในการสร้าง User Interfaces (UI) และให้ความสำคัญกับการนำ UI กลับมาใช้ใหม่ หรือรูปแบบการพัฒนาโปรแกรมแบบเว็บแอปพลิเคชันสามารถสนับสนุนการพัฒนาโปรแกรมในเชิงวัตถุแล้วเขียนโค้ด และ ส่วนของที่สนับสนุนการพัฒนาเครื่องมือสนับสนุนโดยเข้ากันได้กับ HTML5

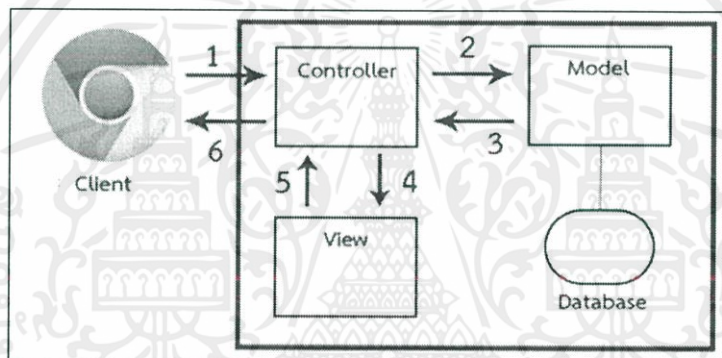
### 2.1.3 ขั้นตอนการทำงานของ Model-View-Controller

MVC คือ สถาปัตยกรรมซอฟต์แวร์ชนิดหนึ่ง ซึ่งปัจจุบันมี Frameworks สำหรับสร้างเว็บแอปพลิเคชันจำนวนมาก แต่เกือบทั้งหมดมีโครงสร้างแบบ MVC ซึ่งแต่ละเว็บแอปพลิเคชันจะถูกแบ่งออกเป็นสามส่วน คือ

1. Model คือ ส่วนที่ใช้ในการติดต่อกับฐานข้อมูล ซึ่งทำหน้าที่ในการดึงข้อมูลขึ้นมาจากฐานข้อมูลขึ้นมา และทำการจัดการข้อมูลต่างๆไว้ในรูปแบบที่เหมาะสม
2. View คือส่วนที่จะนำข้อมูลจาก Model ไปใช้แสดงผลให้ผู้ใช้เห็นผลลัพธ์ออกมาใน User Interface
3. Controller คือส่วนที่จะคอยรับข้อมูลจากลูกค้าเข้ามาจากนั้นนำคำสั่งไปประมวลผลเพื่อสั่งงาน View และ Model ให้ประมวลผลออกมาเป็นอย่างไร

ขั้นตอนการทำงาน MVC แสดงดังรูปที่ 2.1 มีรายละเอียดดังนี้

1. เริ่มจากลูกค้า ส่งข้อมูลไปที่เว็บแอปพลิเคชัน ซึ่งจะถูส่งต่อไปให้ Controller ทำการตรวจสอบข้อมูลที่มาให้
2. แล้ว Controller จะเรียกระบบงานมาให้ทำงานเพื่อจัดการข้อมูลนั้น
3. Model จะทำการคำนวณและอาจเชื่อมโยงกับฐานข้อมูล เพื่อจัดการกับข้อมูลนั้น แล้วส่งผลลัพธ์กลับไป Controller
4. เมื่อ Controller ได้ผลลัพธ์จาก Model แล้วก็ใช้ผลลัพธ์นั้นส่งต่อไปให้ View ทำงาน
5. View จะสร้าง Page สำหรับแสดงผลลัพธ์นั้น แล้วส่ง Page กลับไปที่ Controller
6. Controller ส่ง Page นั้น กลับไปยังลูกค้า



รูปที่ 2.1 ขั้นตอนการทำงานของ Model-View-Controller

## 2.2 ฐานข้อมูล [2]

ระบบที่รวบรวมข้อมูลไว้ในที่เดียวกัน ซึ่งประกอบไปด้วยแฟ้มข้อมูล (File) ระเบียบ (Record) และเขตข้อมูล (Field) ซึ่งถูกจัดการด้วยระบบเดียวกัน โปรแกรมคอมพิวเตอร์จะเข้าไปดึงข้อมูลที่ต้องการได้อย่างรวดเร็ว ซึ่งอาจเปรียบฐานข้อมูลเสมือนเป็น Electronic Filing System ดังหัวข้อต่อไปนี้

### 2.2.1 ฐานข้อมูลและคลังข้อมูล

ก่อนที่จะมีการนำคอมพิวเตอร์มาใช้ในการจัดเก็บข้อมูล การรวบรวมและจัดเก็บข้อมูลจะอยู่ในรูปของแฟ้มเอกสาร มีการจัดหมวดหมู่และจัดแฟ้มข้อมูลเก็บในตู้เอกสาร เมื่อข้อมูลมีมากขึ้นทำให้เกิดปัญหาต้องใช้แฟ้มเอกสารหลายแฟ้ม ทำให้การค้นหาข้อมูลเหล่านี้มีความล่าช้า ต่อมาจึงมีการนำคอมพิวเตอร์มาช่วยในการจัดเก็บข้อมูลจำนวนมากมายทำให้ทำการค้นหาได้เร็วขึ้น โดย ผศ.ดร. ศรีไพโร ศักดิ์รุ่งพงศากุล และ

เจษฎาพร ยุทธนวิบูลย์ชัย (2549) กล่าวถึงฐานข้อมูล (Database) และคลังข้อมูล (Data warehouse) เริ่มจากโครงสร้างเพิ่มข้อมูล จากนั้นจะกล่าวถึงแนวทางในการใช้ฐานข้อมูลการบริหารจัดการข้อมูลองค์ประกอบของฐานข้อมูล รูปแบบของฐานข้อมูล คลังข้อมูลรวมถึงความแตกต่างระหว่างคลังข้อมูลกับฐานข้อมูล ปฏิบัติงาน และรูปแบบการประยุกต์คลังข้อมูลในธุรกิจ

### 2.2.1.1 โครงสร้างข้อมูล

โครงสร้างข้อมูลมีรูปแบบเป็นลำดับชั้นโดยเริ่มด้วยหน่วยที่เล็กที่สุดคือ บิต (Bit) ไบต์(Byte) เขตข้อมูล (Field) ระเบียบข้อมูล (Record) ไฟล์ (File) ตามลำดับ

1. บิต (Bit) เป็นหน่วยที่เล็กที่สุดของข้อมูลที่จัดเก็บในระบบคอมพิวเตอร์
2. ไบต์ (Byte) ประกอบด้วยบิตหลายๆบิตมาเรียงต่อกัน สร้างรหัสแทนข้อมูลเพื่อใช้แทนอักขระ ซึ่งอาจเป็นตัวเลข ตัวอักษร หรือสัญลักษณ์
3. เขตข้อมูล (Field) เป็นการนำข้อมูลหลายอักขระมารวมกันเป็นคำเพื่อให้เกิดความหมาย
4. ระเบียบข้อมูล (Record) กลุ่มของเขตข้อมูลที่มีความสัมพันธ์กันถูกนำมาไว้รวมกัน
5. ไฟล์ (File) กลุ่มของระเบียบข้อมูลที่มีความสัมพันธ์กันถูกนำมาจัดเก็บไว้ด้วยกัน

### 2.2.1.2 ปัญหาเกี่ยวกับเพิ่มข้อมูล

องค์กรโดยส่วนใหญ่มีการพัฒนาระบบสารสนเทศขึ้นมาใช้งานในแต่ละส่วนงาน โดยแต่ละระบบต่างมีไฟล์ข้อมูลเป็นของตนเองโดยไฟล์ที่สร้างขึ้นอาจสร้างมาจากภาษาโปรแกรมที่ต่างกัน เมื่อมีการใช้งานเพิ่มข้อมูลมาถึงระดับหนึ่งทำให้ทราบว่ากรณีที่แต่ละฝ่ายต่างจัดทำโปรแกรมและสร้างเพิ่มข้อมูลขึ้นใช้เฉพาะฝ่ายตนนั้น ส่งผลให้มีการจัดเก็บข้อมูลบางส่วนซ้ำซ้อนกัน หากมีการแก้ไขข้อมูลที่เก็บซ้ำซ้อนกันเหล่านี้แต่แก้ไขไม่ครบจะทำให้ข้อมูลมีค่าไม่สอดคล้องกัน ซึ่งปัญหาความผูกพันระหว่างข้อมูลและโปรแกรม การไม่สามารถใช้ข้อมูลร่วมกันได้ การขาดความคล่องตัว และการการระบบรักษาความปลอดภัยที่ดี

#### 1. ความซ้ำซ้อนของข้อมูล (Data Redundancy)

การจัดเก็บข้อมูลมากกว่าหนึ่งแห่ง ทำให้ยากที่จะควบคุมความถูกต้องตรงกันของข้อมูล เมื่อมีการแก้ไขข้อมูลเกิดขึ้นจะต้องทำการแก้ไขข้อมูลให้ครบทุกแห่งมิเช่นนั้นจะก่อให้เกิดความไม่สอดคล้องกันของข้อมูล ความซ้ำซ้อนของข้อมูลจึงมีค่าใช้จ่าย เปลืองเนื้อที่ในการจัดเก็บ ข้อมูล

#### 2. ความผูกพันระหว่างข้อมูลและโปรแกรม (Program-Data Dependence)

ความไม่เป็นอิสระของข้อมูล (Data Dependence) เป็นความผูกพันระหว่างข้อมูลและโปรแกรมซึ่งเป็นลักษณะของการเขียนโปรแกรมที่ใส่เทคนิคการจัดเก็บ และการเรียกใช้ข้อมูลไว้ใน

โปรแกรม หากมีการเปลี่ยนแปลงโครงสร้างหรือวิธีการเรียกใช้ข้อมูล ย่อมมีผลกระทบต่อโปรแกรม ทำให้ต้องตามแก้ไขโปรแกรมต่างๆ ที่เกี่ยวข้องก่อให้เกิดค่าใช้จ่ายสูงในการบำรุงรักษาโปรแกรม

### 3. การไม่สามารถใช้ข้อมูลร่วมกันได้ (Lack of Data Sharing)

ข้อมูลที่มีการจัดเก็บแยกจากกันโดยแต่ละส่วนงานต่างจัดเก็บข้อมูลเป็นของตนเองทำให้ความพร้อมของการใช้ข้อมูลอยู่ระดับต่ำเนื่องจากเป็นการยากหรือไม่สามารถนำข้อมูลจากหลายแฟ้มมาใช้งานร่วมกันได้

### 4. การขาดความคล่องตัว (Lack of Flexibility)

ระบบแฟ้มข้อมูลขาดความคล่องตัวในการตอบสนองต่อความต้องการใหม่ๆ เนื่องจากแฟ้มข้อมูลไม่สนับสนุนงานในรูปแบบที่ไม่เคยทำเป็นประจำ

### 5. การขาดระบบรักษาความปลอดภัยที่ดี (Poor Security)

ป้องกันไม่ให้ผู้ที่มีสิทธิเข้าถึงข้อมูลและใช้ข้อมูล รวมทั้งการกำหนดว่าผู้ใช้รายใดสามารถเข้าถึงข้อมูลได้ในระดับใดได้บ้างจะทำได้ยากโดยการเขียนโปรแกรม วิธีรักษาความปลอดภัยของระบบแฟ้มข้อมูลมีขอบเขตความสามารถค่อนข้างจำกัด

## 2.2.1.3 แนวทางในการใช้ฐานข้อมูลในการบริหารจัดการข้อมูล

### 1. ลดความซ้ำซ้อนของข้อมูล (Minimum Redundancy)

การจัดเก็บข้อมูลโดยระบบไฟล์นั้น แต่ละส่วนงานจะมีไฟล์ที่ใช้เฉพาะในส่วนงานของตน ส่งผลให้จัดเก็บข้อมูลเดียวกันไว้หลายแห่ง การนำข้อมูลมารวมกันเพื่อลดส่วนที่ซ้ำซ้อนของข้อมูลได้ในกรณีที่มีความจำเป็นทางเทคนิคที่จะต้องเก็บข้อมูลชุดเดียวกันมากกว่าหนึ่งแห่ง ระบบฐานข้อมูลจะมีระบบจัดการฐานข้อมูล ซึ่งเป็นซอฟต์แวร์ที่ดูแลจัดการข้อมูลทำให้สามารถควบคุมการเกิดความซ้ำซ้อนของข้อมูลได้

### 2. มีความเป็นอิสระของข้อมูล (Data Independence)

ในระบบฐานข้อมูลจะมีแหล่งรวมของข้อมูล ที่เก็บรายละเอียดของข้อมูลต่างๆ ไว้ที่ส่วนกลาง และมี ระบบจัดการฐานข้อมูล(DBMS) ทำหน้าที่ในการดูแลการปรับเปลี่ยนโครงสร้างของข้อมูล ทำให้โปรแกรมต่างๆนี้เป็นอิสระจากการเปลี่ยนแปลงโครงสร้างของข้อมูล

### 3. สนับสนุนการใช้ข้อมูลร่วมกัน (Improved Data Sharing)

ผู้ใช้งานภายในองค์กรต้องการใช้ข้อมูลที่แตกต่างกัน และอาจต้องการข้อมูลบางส่วนที่เหมือนกัน การจัดเก็บข้อมูลไว้ที่ส่วนกลางช่วยให้สามารถใช้ข้อมูลร่วมกันได้โปรแกรมประยุกต์ที่จะพัฒนาขึ้นใหม่สามารถใช้ข้อมูลที่มีอยู่แล้วได้โดยไม่ต้องเพิ่มข้อมูลเข้าไปในระบบอีก

#### 4. มีความคล่องตัวในการใช้งาน (Improved Flexibility)

การรวบรวมข้อมูลไว้ที่เดียวกัน และมีการควบคุมอยู่ที่ส่วนกลางจะช่วยให้มีความคล่องตัวในการใช้งานมากกว่าระบบไฟล์ข้อมูล หากข้อมูลที่ต้องการได้รับการออกแบบและพัฒนาไว้แล้วผู้เขียนโปรแกรมสามารถพัฒนาโปรแกรมใหม่ได้โดยไม่ต้องเสียเวลาในการออกแบบข้อมูลใหม่

#### 5. มีระบบรักษาความปลอดภัยของข้อมูลสูง (High Degree of Data Integrity)

ฐานข้อมูลจะมีระบบรักษาความปลอดภัยของข้อมูลโดย DBMS จะตรวจสอบรหัสผ่านเข้าสู่ระบบและจะอนุญาตให้ผู้ใช้ที่มีสิทธิเข้ามาในระบบทำการเรียกดูข้อมูลหรือแก้ไขข้อมูลได้เฉพาะสิทธิที่กำหนดให้แต่ละคนเท่านั้น

### 2.2.2 Database Management System

DBMS คือ ระบบการจัดการฐานข้อมูล โดยอำนวยความสะดวกให้แก่ผู้ใช้ทั้งในด้านการสร้าง การปรับปรุงแก้ไข การเข้าถึงข้อมูล และการจัดการเกี่ยวกับระบบเพิ่มข้อมูลทางกายภาพภายในฐานข้อมูลซึ่งต่างไปจากระบบเพิ่มข้อมูล คือ หน้าที่เหล่านี้จะเป็นของโปรแกรมเมอร์ ในการติดต่อฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่ม DML หรือ DDL หรือ จะด้วยโปรแกรมต่างๆ ทุกคำสั่งที่ใช้กระทำกับฐานข้อมูลจะถูกโปรแกรม DBMS นำมาแปล เป็นการกระทำต่างๆภายใต้คำสั่งนั้น เพื่อนำไปกระทำกับตัวข้อมูลใน ฐานข้อมูลต่อไป

DBMS ถูกพัฒนาขึ้นเพื่อแก้ไขปัญหาด้าน Data Independence ที่ไม่มีในระบบเพิ่มข้อมูล ทำให้มีความเป็นอิสระจากทั้งส่วนของฮาร์ดแวร์ และข้อมูลภายในฐานข้อมูลกล่าวคือโปรแกรม DBMS นี้จะมีการทำงานที่ไม่ขึ้นอยู่กับรูปแบบ (Platform) ของตัวฮาร์ดแวร์ ที่นำมาใช้กับระบบฐานข้อมูลรวมทั้งมีรูปแบบในการอ้างอิงข้อมูลที่ไม่ขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูลด้วยการใช้ Query Language ส่งผลให้ผู้ใช้สามารถเรียกใช้ข้อมูลจากฐานข้อมูลได้โดยไม่ต้องทราบถึงประเภทหรือขนาดของข้อมูลนั้นหรือสามารถกำหนดลำดับที่ของฟิลด์ ในการกำหนดการแสดงผลได้โดยไม่ต้องคำนึงถึงลำดับที่จริงของฟิลด์นั้น

#### 2.2.2.1 หน้าที่ของ Database Management System

1. ทำหน้าที่แปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูลให้อยู่ในรูปแบบที่เข้าใจ
2. ทำหน้าที่ในการนำคำสั่งต่างๆ ซึ่งได้รับการแปลแล้วไปส่งให้ฐานข้อมูลทำงาน เช่น การเรียกใช้ข้อมูล (Retrieve) การจัดเก็บข้อมูล (Update) การลบข้อมูล (Delete) หรือ การเพิ่มข้อมูลเป็นต้น (Add)
3. ทำหน้าที่ป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะคอยตรวจสอบว่าคำสั่งใดที่สามารถทำงานได้และคำสั่งใดที่ไม่สามารถทำได้
4. ทำหน้าที่รักษาความสัมพันธ์ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ

5. ทำหน้าที่เก็บรายละเอียดต่างๆ ที่เกี่ยวข้องกับข้อมูลภายในฐานข้อมูลไว้ใน Data Dictionary ซึ่งรายละเอียดเหล่านี้มักจะถูกเรียกว่า ข้อมูลของข้อมูล (Meta Data)

6. ทำหน้าที่ควบคุมให้ฐานข้อมูลทำงานได้อย่างถูกต้องและมีประสิทธิภาพ

#### 2.2.2.2 การจัดการฐานข้อมูล

การจัดการฐานข้อมูล (Database Management) คือ การบริหารแหล่งข้อมูลที่ถูกเก็บรวบรวมไว้ที่ศูนย์กลาง เพื่อตอบสนองต่อการใช้ของโปรแกรมประยุกต์อย่างมีประสิทธิภาพและลดการซ้ำซ้อนของข้อมูล รวมทั้งความขัดแย้งของข้อมูลที่เกิดขึ้นภายในองค์กร ในอดีตการเก็บข้อมูลมักจะเป็นอิสระต่อกันไม่มีการเชื่อมโยงของข้อมูลเกิดการ สิ้นเปลืองพื้นที่ในการเก็บข้อมูล ดังรูปที่ 2.2

#### 2.2.2.3 ส่วนประกอบที่สำคัญในระบบการจัดการข้อมูล

ระบบการจัดการฐานข้อมูล จะมีส่วนประกอบที่สำคัญ 3 ส่วนได้แก่

##### 1. ภาษาคำนิยามของข้อมูล (Data Definition Language)

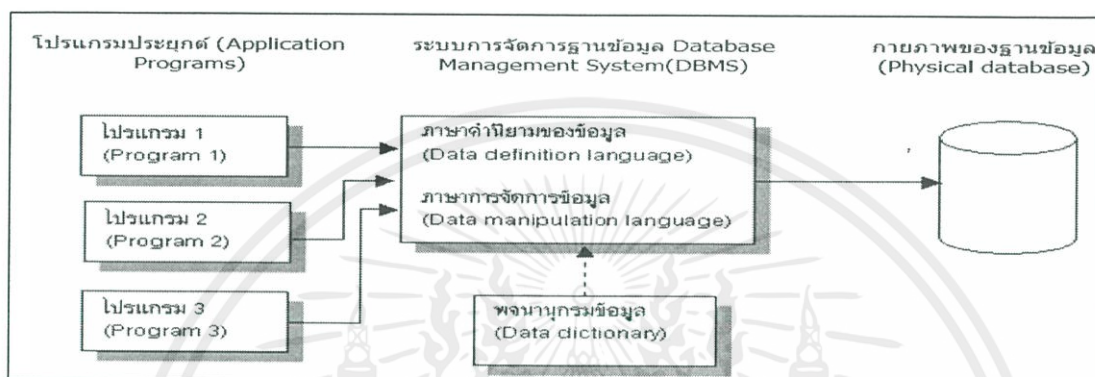
ในส่วนนี้จะกล่าวถึงส่วนประกอบของระบบการจัดการฐานข้อมูลว่าข้อมูลแต่ละส่วนประกอบด้วยอะไรบ้าง (Data Element) ในฐานข้อมูลซึ่งเป็นภาษาทางการที่นักเขียนโปรแกรมใช้ในการสร้างเนื้อหาข้อมูล และโครงสร้างข้อมูลก่อนที่ข้อมูลดังกล่าวจะถูกแปลงเป็นแบบฟอร์มที่ต้องการของโปรแกรมประยุกต์ จะประกอบด้วยคำสั่งที่ใช้ในการกำหนดโครงสร้างข้อมูลว่ามีคอลัมน์อะไร แต่ละคอลัมน์เก็บข้อมูลประเภทใด รวมถึงการเพิ่มคอลัมน์ การ กำหนดดัชนี

##### 2. ภาษาการจัดการฐานข้อมูล (Data Manipulation Language)

เป็นภาษาเฉพาะที่ใช้ในการจัดการระบบฐานข้อมูล ซึ่งอาจจะเป็นการเชื่อมโปรแกรมภาษาในยุคที่สามและยุคที่สี่เข้าด้วยกันเพื่อจัดการข้อมูลในฐานข้อมูล ภาษานี้มักจะประกอบด้วยคำสั่งที่อนุญาตให้ผู้ใช้สามารถสร้างโปรแกรมพิเศษขึ้นมา รวมถึงข้อมูลต่างๆ ในปัจจุบันที่นิยมใช้ ได้แก่ ภาษา SQL (Structure Query Language)

### 3. พจนานุกรมข้อมูล (Data Dictionary)

เป็นเครื่องมือสำหรับการเก็บและการจัดข้อมูลสำหรับการบำรุงรักษาในฐานข้อมูล โดยพจนานุกรมจะมีการกำหนดชื่อของสิ่งต่างๆ (Entity) และระบุไว้ในโปรแกรมฐานข้อมูล



รูปที่ 2.2 ส่วนประกอบของระบบฐานข้อมูลแสดงส่วนประกอบของระบบจัดการฐานข้อมูล

#### 2.2.2.4 องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูลประกอบด้วยองค์ประกอบหลัก 4 ส่วน คือ

1. ข้อมูล (Data) หมายถึง ข้อมูลและความสัมพันธ์ของข้อมูลที่จัดเก็บอยู่ในฐานข้อมูล
2. ฮาร์ดแวร์ (Hardware) ได้แก่ เครื่องคอมพิวเตอร์และอุปกรณ์รอบข้าง (Peripherals)
3. ซอฟต์แวร์ (Software) ได้แก่ ระบบปฏิบัติการ (Operating Systems) และระบบจัดการข้อมูล (Database Management System) รวมทั้งโปรแกรมยูทิลิตี้ต่าง ๆ
4. ผู้ใช้ (Users) ได้แก่ บุคลากรที่เกี่ยวข้องกับระบบฐานข้อมูล เช่น ผู้บริหารฐานข้อมูล (Database Administrator) นักวิเคราะห์ระบบ (System Analysts) ผู้เขียนโปรแกรมประยุกต์ (Programmers) และผู้ใช้งาน (End Users) เป็นต้น

#### 2.2.2.5 อุปสรรคในการพัฒนาระบบฐานข้อมูล

1. ความผิดพลาดจากการป้อนข้อมูลเข้าย่อมมีผลกระทบต่อหน่วยงานอื่นที่นำข้อมูลนั้นไปใช้ เนื่องจากไม่มีข้อมูลอื่นที่มาเทียบกับข้อมูลในฐานข้อมูลชุดนั้น

2. สร้างเพิ่มข้อมูลร่วมเพื่อตอบสนองกับองค์กร ทุกแผนกกระทำได้ยากเนื่องจากแต่ละแผนกอาจจะต้องการได้ข้อมูลในความละเอียดที่ไม่เท่ากันผู้จัดการระดับล่างต้องการใช้ข้อมูลเพื่อการทำงานวันต่อวัน แต่ผู้บริหารระดับสูงต้องการใช้ข้อมูลเพื่อการวางแผน ดังนั้นการออกแบบฐานข้อมูลจึงทำได้ยากมาก

3. ในเรื่องของความปลอดภัยทั้งนี้เนื่องจากทุกแผนกมีการใช้ข้อมูลร่วมกันจึงต้องมีการสร้างระบบป้องกันความปลอดภัยของข้อมูลเพื่อไม่ให้ข้อมูลรั่วไหลจะต้องมีการกำหนดรหัสผ่าน (Password) และการจัดลำดับความสำคัญของงาน (Priority)

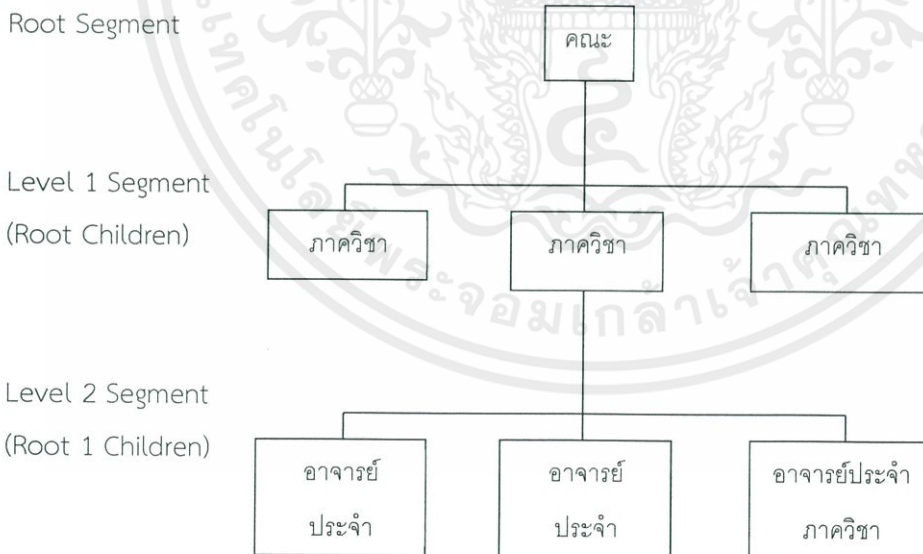
### 2.2.3 การออกแบบฐานข้อมูล

#### 2.2.3.1 รูปแบบของฐานข้อมูล (Database Model)

ผศ.ดร. ศรีไพโร ศักดิ์รุ่งพงศากุล และ เจษฎาพร ยุทธนวิบูลย์ชัย (2549) ได้ให้รายละเอียดเกี่ยวกับประเภทของฐานข้อมูลดังนี้ ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database Model) ฐานข้อมูลแบบเครือข่าย (Network Database Model) และฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database Model)

##### 1. แบบจำลองฐานข้อมูลลำดับชั้น (Hierarchical Database Model)

มีโครงสร้างคล้ายโครงสร้างของต้นไม้ (Tree Structure) ข้อมูลมีความสัมพันธ์กันในลักษณะ One-to-Many ข้อมูลจะได้รับการจัดเก็บในรูปแบบของ Segment โดย Segment ที่อยู่บนสุดเรียกว่า Root Node ถัดลงมาเรียกว่า Child Node ดังแสดงในรูปที่ 2.3

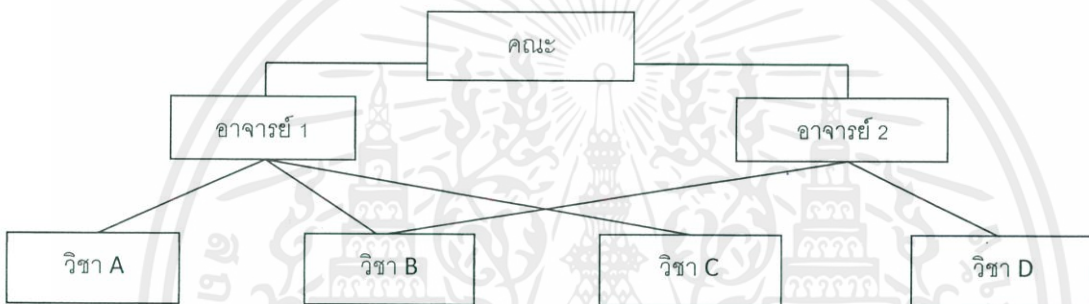


รูปที่ 2.3 แบบจำลองฐานข้อมูลลำดับชั้น

ข้อดีและข้อจำกัดของแบบจำลองฐานข้อมูลลำดับชั้น คือ เป็นแบบจำลองฐานข้อมูลลำดับชั้นที่มีโครงสร้างที่เข้าใจง่าย และเหมาะกับข้อมูลที่มีการเรียงลำดับอย่างต่อเนื่อง แต่ไม่สามารถรองรับความสัมพันธ์ของข้อมูลในลักษณะ Many-to-Many และการเข้าถึงข้อมูลจะมีความคล่องตัวน้อย

## 2. แบบจำลองฐานข้อมูลเครือข่าย (Network Database Model)

โครงสร้างของข้อมูลที่น่าเสนอเป็นลักษณะ Multi-List Structure โดยมีความสัมพันธ์ของข้อมูลเป็นแบบ Many-to-Many จะต่างจากแบบจำลองฐานข้อมูลลำดับชั้นตรงที่แต่ละลำดับชั้นตรงที่แต่ละ Segment สามารถมี Parent ได้มากกว่าหนึ่ง และจะเรียกว่า Parent ว่า Owner ส่วน Child จะเรียกว่า Member ดังแสดงในรูปที่ 2.4

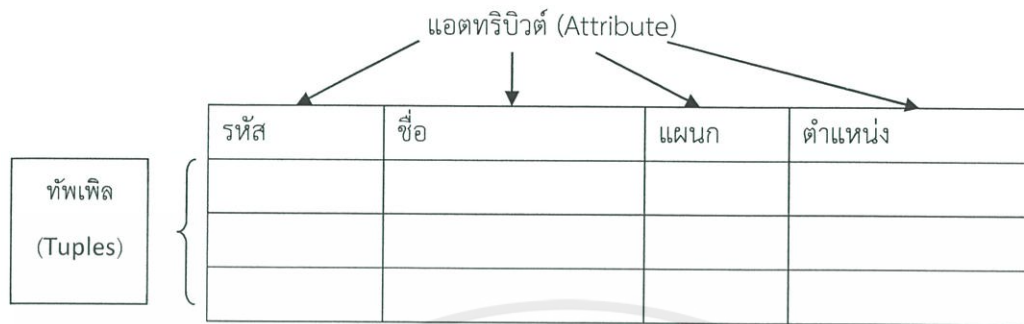


รูปที่ 2.4 แบบจำลองฐานข้อมูลเครือข่าย

ข้อดีและข้อจำกัดของแบบจำลองฐานข้อมูลเครือข่าย คือ ความซับซ้อนของข้อมูลมีน้อยกว่าแบบจำลองฐานข้อมูลลำดับชั้น และสนับสนุนความสัมพันธ์ของข้อมูลในลักษณะ Many-to-Many ซึ่งสามเชื่อมโยงข้อมูลแบบไป-กลับได้ โดยจะใช้พอยเตอร์ในการเชื่อมโยงและเข้าถึงข้อมูล แต่วิธีนี้จะเปลืองเนื้อที่ในการจัดเก็บพอยเตอร์ และยังมีคามยุ่งยากอยู่ ซึ่งเป็นการเปลี่ยนแปลงโครงสร้างข้อมูลที่มีความซับซ้อน อีกทั้งผู้เขียนโปรแกรมจะต้องเข้าใจในโครงสร้างของข้อมูลเป็นอย่างดี

## 3. แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

แบบจำลองฐานข้อมูลเชิงสัมพันธ์แสดงโครงสร้างของข้อมูลในรูปแบบของตารางและจะเรียกรายการว่า รีเลชัน (Relation) โดยแต่ละรีเลชันจะประกอบด้วย แถวหรือทัพเพิล (Tuple) และคอลัมน์ ซึ่งเรียกว่า แอตทริบิวต์ (Attribute) ในแต่ละรีเลชันจะมีแอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ ซึ่งเรียกว่า คีย์ (Key) เป็นตัวบอกถึงความแตกต่างของแต่ละทัพเพิล ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 แบบจำลองฐานข้อมูลเชิงสัมพันธ์

ข้อดีและข้อจำกัดของแบบจำลองฐานข้อมูลเชิงสัมพันธ์ คือ แบบจำลองฐานข้อมูลเชิงสัมพันธ์มีโครงสร้างที่เข้าใจง่าย มีระบบจัดการฐานข้อมูลที่ช่วยให้การจัดการกับข้อมูลทำได้ง่าย ผู้ใช้ไม่จำเป็นต้องรู้ถึงโครงสร้างการจัดเก็บข้อมูลทางกายภาพเพราะจะซ่อนความซับซ้อนของระบบไว้ และข้อมูลมีความเป็นอิสระจากโปรแกรม

### 2.2.3.2 ทฤษฎีวงจรการพัฒนาระบบฐานข้อมูล [3]

โอภาส เอี่ยมสิริวงศ์ (2551) ได้ให้รายละเอียดเกี่ยวกับวงจรการพัฒนาระบบฐานข้อมูลดังนี้ วงจรการพัฒนาระบบฐานข้อมูลแบบ DBLC (Database Life Cycle) เป็นขั้นตอนที่กำหนดขึ้น เพื่อใช้เป็นแนวทางในการพัฒนาระบบฐานข้อมูล ซึ่งวงจรนี้จะทำให้เราทราบถึงกิจกรรมต่างๆในแต่ละระยะ ตั้งแต่การวางแผน การกำหนดระบบ การรวบรวมความต้องการ จนกระทั่งการพัฒนาและการนำไปใช้งาน ซึ่งประกอบด้วยขั้นตอนต่างๆ ดังต่อไปนี้

#### 1. การศึกษาเบื้องต้นเพื่อจัดทำฐานข้อมูล (Database Initial Study)

การศึกษาเบื้องต้นมีวัตถุประสงค์เพื่อทำความเข้าใจโครงสร้างการทำงาน ปัญหาการทำงาน และขอบเขตของการจัดทำฐานข้อมูล ประกอบด้วย

1. การวิเคราะห์ จะทำการศึกษาสภาพการทำงาน วัตถุประสงค์ และโครงสร้าง
2. การกำหนดปัญหาและเงื่อนไข เป็นการศึกษาปัญหาจากการปฏิบัติงาน
3. การกำหนดวัตถุประสงค์ เป็นการกำหนดวัตถุประสงค์การจัดทำ และผู้ที่จะใช้ฐานข้อมูล
4. การกำหนดขอบเขต เป็นการกำหนดขอบเขตของการจัดทำฐานข้อมูล

## 2. การออกแบบฐานข้อมูล (Database Design)

เป็นขั้นตอนที่นำเอารายละเอียดต่างๆ ที่ได้จากการวิเคราะห์ในขั้นตอนแรกมากำหนดเป็นแนวทางในการออกแบบฐานข้อมูล ซึ่งแบ่งออกเป็น 3 ระดับคือ

1. ระดับแนวความคิด (Conceptual Database Design) เป็นการกำหนดโครงสร้างของฐานข้อมูล มีจุดมุ่งหมายเพื่ออธิบายโครงสร้างหลักๆ ของข้อมูลภายในระบบฐานข้อมูล การออกแบบในระดับนี้จะสามารถอธิบายข้อมูล (Entities) อยู่ในฐานข้อมูลได้ มีการจัดเก็บรายละเอียดข้อมูล (Attributes) และมีความสัมพันธ์ (Relations) ระหว่างข้อมูลเหล่านั้น โดยผลของการออกแบบในระดับนี้เป็นรูปแบบจำลองของข้อมูลที่ประกอบด้วยโครงสร้างที่ยังไม่สามารถนำไปใช้ได้จริง

2. ระดับตรรกะ (Logical Database Design) เป็นการนำโครงสร้างที่ได้จากระดับความคิดมาตรวจสอบความถูกต้องขึ้นกับส่วนประมวลผลต่างๆ ที่ออกแบบไว้ และปรับปรุงให้เป็นไปตามโครงสร้างข้อมูลฐานข้อมูลที่จะนำไปใช้งาน ขั้นตอนการออกแบบระดับนี้เน้นความสำคัญในส่วนของ การจัดกลุ่มข้อมูลโดยไม่เกิดความซ้ำซ้อนด้วยวิธีการทำให้เป็นรูปแบบที่เป็นบรรทัดฐาน (Normalization) เพื่อปรับการออกแบบฐานข้อมูลให้เหมาะสม

3. ระดับกายภาพ (Physical Database Design) เป็นขั้นตอนที่กำหนดข้อมูลที่จัดเก็บลงฐานข้อมูลจริง มีการกำหนดวิธีการเข้าถึงข้อมูล (Access Method) ประเภทของข้อมูล (Data Type) โครงสร้างข้อมูล (Data Structure) การจัดระเบียบแฟ้ม (File Organization) เป็นต้น โดยการออกแบบฐานข้อมูลระดับนี้จะสามารถนำไปใช้ในการสร้างฐานข้อมูลจริง ทั้งนี้ต้องเลือกที่จะใช้โปรแกรมหรือซอฟต์แวร์เพื่อช่วยจัดการข้อมูล หรือรายงานต่างๆ

## 3. การสร้างฐานข้อมูล (Implementation and Loading)

เป็นขั้นตอนนำโครงสร้างต่างๆ ของระบบฐานข้อมูลที่ได้ออกแบบมาสร้างเป็นฐานข้อมูลที่ใช้เก็บข้อมูลจริง รวมทั้งการแปลงข้อมูลจากระบบงานเดิมในกรณีที่ใช้ระบบคอมพิวเตอร์ในการประมวลผลอยู่แล้วให้สามารถใช้ในระบบฐานข้อมูลที่พัฒนาขึ้นใหม่ได้ การจัดทำและนำข้อมูลเข้าสู่ฐานข้อมูล

## 4. การทดสอบระบบ (Testing and Evaluation)

เป็นขั้นตอนในการทดสอบเพื่อหาข้อผิดพลาดต่างๆ รวมถึงประเมินความสามารถของระบบฐานข้อมูล เพื่อนำไปใช้เป็นแนวทางการปรับปรุงระบบฐานข้อมูลให้พัฒนาขึ้น

## 5. การนำไปใช้ (Operation)

เป็นขั้นตอนนำระบบฐานข้อมูลที่พัฒนาเสร็จสมบูรณ์แล้วไปใช้งานจริง โดยจัดให้มีการฝึกอบรมแก่ผู้ใช้งาน เพื่อให้เข้าใจการทำงานของระบบและสามารถใช้โปรแกรมได้โดยไม่มีปัญหา

## 6. การบำรุงรักษาและการประเมินระบบ (Maintenance and Evolution)

เป็นการบำรุงรักษาเพื่อให้ระบบฐานข้อมูลทำงานได้อย่างมีประสิทธิภาพ รวมทั้งเป็นขั้นตอนการแก้ไข และปรับปรุงระบบฐานข้อมูลกรณีที่มีการเพิ่มหรือเปลี่ยนแปลงความต้องการของผู้ใช้ที่ส่งผลกระทบต่อระบบฐานข้อมูล

โดยสรุป วงจรพัฒนาระบบมีวัตถุประสงค์เพื่อใช้เป็นแนวสร้างพัฒนาระบบฐานข้อมูล ประกอบด้วย 6 ขั้นตอน ซึ่งมีการทำงานเรียงลำดับแต่ละขั้นตอนและวนกลับเป็นวงจร ได้แก่ การวิเคราะห์ความต้องการของผู้ใช้ การออกแบบฐานข้อมูล การสร้างฐานข้อมูล การทดสอบระบบ การนำไปใช้ และการบำรุงรักษาและประเมินผล

### 2.2.4 เครื่องมือที่ใช้ในการวิเคราะห์ระบบ









#### 2.2.4.1 แผนผังการไหลของข้อมูล Flow Chart

ผังงาน คือ แผนภาพที่มีการใช้สัญลักษณ์รูปภาพและลูกศรที่แสดงถึงขั้นตอนการทำงานของโปรแกรมหรือระบบทีละขั้นตอน รวมไปถึงทิศทางการไหลของข้อมูลตั้งแต่แรกจนได้ผลลัพธ์ตามที่ต้องการการเขียนผังงานสามารถแบ่งออกได้เป็นสองประเภทใหญ่ๆ คือ





1. ผังงานระบบ (System Flow Chart) ใช้แสดงขั้นตอนการทำงานภายในระบบงานหนึ่งๆ โดยกล่าวถึงข้อมูลต่างๆ ที่เกี่ยวข้องทั้งหมด เช่น เอกสารเบื้องต้นคืออะไร วัสดุที่ใช้คืออะไร ใช้หน่วยความจำประเภทใด จะต้องส่งผ่านไปยังหน่วยงานใด วิธีการประมวลผลและการแสดงผลลัพธ์

2. ผังงานโปรแกรม (Program Flow Chart) การเขียนผังโปรแกรมจะประกอบไปด้วยการใช้สัญลักษณ์มาตรฐานต่าง ๆ ที่เรียกว่า สัญลักษณ์ ANSI (American National Standards Institute) ในการสร้างผังงาน ดังตัวอย่างที่แสดงในตารางที่ 2.1

ตารางที่ 2.1 ความหมายของสัญลักษณ์ Flow Chart

สัญลักษณ์	ความหมาย
	Terminator ใช้แสดงจุดเริ่มต้นและจุดสิ้นสุดของโปรแกรม
	Process ใช้ในการประมวลผลข้อมูล กำหนดค่า หรือการคำนวณทางคณิตศาสตร์
	Input/Output หรือ I/O ใช้ในการรับข้อมูล แสดงผลข้อมูลโดยไม่วะบุอุปกรณ์
	Manual Input ใช้ในการรับข้อมูลจากแป้นพิมพ์
	Decision Symbol ใช้ในการเปรียบเทียบเงื่อนไขหรือตัดสินใจ
	Display ใช้ในการเปรียบเทียบเงื่อนไขหรือตัดสินใจ
	Document Symbol ใช้เมื่อต้องการระบุให้แสดงข้อมูลบนเครื่องพิมพ์
	Preparation การเตรียมทำงานลำดับขั้นต่อไป

ตารางที่ 2.1 ความหมายของสัญลักษณ์ Flow Chart (ต่อ)

สัญลักษณ์	ความหมาย
	Predefined Process โปรแกรมย่อย หรือโมดูลเริ่มทำงานหลังจากจบคำสั่งในโปรแกรมย่อยแล้ว จะกลับมาทำคำสั่งต่อไป
	Online Storage แหล่งเก็บข้อมูลออนไลน์ หรือหน่วยความจำสำรอง
	Connector หรือ On-page Connector จุดเชื่อมผังงานในหน้าเดียวกัน
	Connector หรือ Off-page Connector จุดเชื่อมผังงานในหน้าต่างกัน

2.2.4.2 อีอาร์โมเดล

1. เอ็นทิตี

เอ็นทิตี หมายถึง บุคคล สถานที่ เหตุการณ์ หรือสิ่งต่างๆ ที่จำเป็นต้องนำมาจัดเก็บ เช่น ถ้าต้องการจัดทำฐานข้อมูลของมหาวิทยาลัย สิ่งที่เป็นเอ็นทิตี ได้แก่ นักศึกษา อาจารย์ วิชา อาคารเรียน เป็นต้น

2. แอททริบิวต์

ในแต่ละเอ็นทิตี ประกอบด้วย คุณลักษณะ เรียกว่า แอททริบิวต์ เช่น เอ็นทิตีนักศึกษา ประกอบด้วยแอททริบิวต์ รหัสนักศึกษา ชื่อ นามสกุล เกรด วันเดือนปีเกิด เพศ ที่อยู่ หมายเลขโทรศัพท์ สาขาวิชาเอก หรือเอ็นทิตีเครื่องบิน ประกอบด้วยแอททริบิวต์ ดังนี้ หมายเลขเครื่องบิน วันที่บำรุงรักษาเครื่องล่าสุด จำนวนชั่วโมงบิน จำนวนชั่วโมงบินหลังจากบำรุงรักษาครั้งล่าสุด ดังนั้นจะเห็นว่าแอททริบิวต์เหล่านี้มีประโยชน์ที่จะนำไปจัดเก็บ เพื่อนำไปประมวลผลต่อไป

### 3. ตารางและคุณลักษณะของตาราง

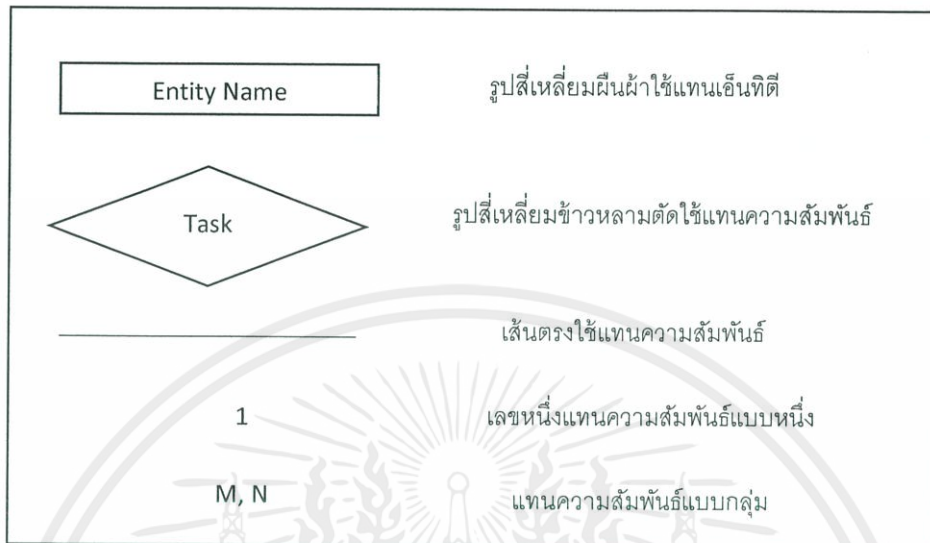
ในฐานข้อมูลหนึ่งจะมีเอ็นทิตีได้หลายเอ็นทิตี โดยแต่ละเอ็นทิตีมีความสัมพันธ์กัน โครงสร้างลักษณะแบบนี้เรียกว่า ตาราง ทั้งนี้เพื่อเหตุผลในการแก้ปัญหาความซ้ำซ้อนของข้อมูล และความไม่สอดคล้องของข้อมูลแสดงไว้ในตารางที่ 2.2

ตารางที่ 2.2 คุณลักษณะของตารางความสัมพันธ์

1.	ตารางต้องมีลักษณะ 2 มิติประกอบด้วยแถวและสดมภ์
2.	ในแต่ละแถวแทนค่าข้อมูลเพียง 1 เอ็นทิตีเซต
3.	ในแต่ละคอลัมน์แทนด้วยแอทริบิวต์และในแต่ละคอลัมน์มีชื่อจำเพาะ
4.	แต่ละแถวกับคอลัมน์ที่ตัดกันแทนข้อมูลเพียง 1 ค่าเท่านั้น
5.	แต่ละตารางต้องมีแอทริบิวต์ที่เป็นคีย์หลักที่บอกความเป็นเอกลักษณ์ของแถวได้
6.	ข้อมูลที่เก็บในตารางต้องมีรูปแบบเดียวกัน เช่น ข้อมูลที่เก็บวันเดือนปีเกิด ต้องเก็บข้อมูลเป็น Date โดยมีลักษณะเป็นตัวขึ้นด้วยสัญลักษณ์ '-'
7.	ในแต่ละคอลัมน์มีการเก็บขนาดของข้อมูลบอกไว้ เรียกว่า แอทริบิวต์โดเมน
8.	ในแต่ละแถวบอกรายละเอียดสารสนเทศ ได้ 1 อย่าง
9.	ลำดับการนำเข้าสู่ฐานข้อมูลเป็นหน้าที่ของ DBMS ไม่มีผลต่อผู้ใช้

#### 2.2.4.3 ความสัมพันธ์ในฐานข้อมูลเชิงสัมพันธ์

พื้นฐานความสัมพันธ์ของตารางข้อมูล ซึ่งมีทั้งหมด 3 ชนิดด้วยกันคือ ความสัมพันธ์ชนิดหนึ่งต่อกลุ่ม (1:M) กลุ่มต่อกลุ่ม (M : N) และหนึ่งต่อหนึ่ง (1:1) ความสัมพันธ์ที่พบและใช้มากที่สุดได้แก่ ความสัมพันธ์ชนิดหนึ่งต่อกลุ่ม และกลุ่มต่อกลุ่ม สำหรับความสัมพันธ์ชนิดหนึ่งต่อหนึ่งนั้นมีการนำไปใช้งาน เช่น ในการบริหารงานมหาวิทยาลัย จะมีอธิการบดีเพียง 1 คนทำหน้าที่บริหารมหาวิทยาลัยได้เพียง 1 แห่งเท่านั้น เช่นเดียวกับ คณบดีทำหน้าที่บริหารงานคณะใดคณะหนึ่งเพียงหนึ่งคณะเท่านั้น เป็นต้น ดังนั้นจึงเขียนความสัมพันธ์ของทั้งสองเอ็นทิตีได้เป็น 1 : 1 แต่วิธีการสื่อความหมายระหว่างผู้ออกแบบฐานข้อมูลกับโปรแกรมเมอร์จะใช้แผนผังแทนแผนผังนี้แสดงดังรูปที่ 2.6 เรียกว่า อีอาร์ไดอะแกรม สัญลักษณ์ที่ใช้ในการทำแผนผังมีหลายรูปแบบ ซึ่งจะใช้รูปแบบของ เชิน (Chen) มีดังนี้



รูปที่ 2.6 สัญลักษณ์ที่ใช้ในการทำแผนผัง

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ความสัมพันธ์แบบหนึ่งต่อหนึ่ง หมายถึง ข้อมูลที่เก็บอยู่ในเอนทิตีหนึ่งมีความสัมพันธ์กับข้อมูลที่เก็บอยู่ในเอนทิตีอื่นเพียงหนึ่งระเบียนเท่านั้น และสมาชิกของเอนทิตีอื่นมีความสัมพันธ์กับเอนทิตีเดิมเพียงหนึ่งระเบียนด้วย

2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

ความสัมพันธ์แบบหนึ่งต่อกลุ่ม หมายถึง สมาชิกจำนวนหนึ่งระเบียนที่เก็บอยู่ในเอนทิตีหนึ่งมีความสัมพันธ์กับสมาชิกที่เก็บอยู่ในเอนทิตีอื่นจำนวนหลายระเบียน อาจตั้งแต่หนึ่งระเบียนขึ้นไป เพราะฉะนั้นความสัมพันธ์เป็น หนึ่งต่อกลุ่ม สัญลักษณ์ที่ใช้คือ 1 : M

3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ความสัมพันธ์แบบกลุ่มต่อกลุ่ม หมายถึง สมาชิกจำนวนตั้งแต่หนึ่งระเบียนขึ้นไปในเอนทิตีหนึ่งมีความสัมพันธ์กับสมาชิกในเอนทิตีที่สองจำนวนหลายระเบียน ในขณะเดียวกันหากมองย้อนกลับ สมาชิกในเอนทิตีที่สองกลับมีความสัมพันธ์กับสมาชิกในเอนทิตีที่หนึ่งจำนวนหลายระเบียนเช่นกัน

#### 2.2.4.4 คีย์ (Key)

ด้วยปัญหาหลักของการไม่ใช้ฐานข้อมูล ได้แก่ การซ้ำซ้อนของข้อมูล เพราะฉะนั้นการควบคุมไม่ให้มีข้อมูลซ้ำซ้อนจึงต้องมีเครื่องมือมาจัดการ คือ คีย์ คีย์ช่วยในการแบ่งแยกความแตกต่างของข้อมูลในแต่ละแถว และคีย์ยังใช้ในการเชื่อมโยงข้อมูลระหว่างตารางตั้งแต่ 2 ตารางขึ้นไปด้วย นอกจากนี้ยังเป็นเครื่องมือที่ช่วยในการกำหนดความสัมพันธ์ของเอ็นทิตี ประกอบไปด้วย

##### 1. ซูเปอร์คีย์

ซูเปอร์คีย์ (Superkey) หมายถึง แอททริบิวต์หรือกลุ่มของแอททริบิวต์ที่สามารถระบุความเป็นเอกลักษณ์ของแต่ละแถวเพื่อให้แตกต่างกันในตารางได้

##### 2. คีย์คู่แข่ง

คีย์คู่แข่ง (Candidate Key) หมายถึง ซูเปอร์คีย์ที่มีขนาดเล็กที่สุด ที่ไม่มีแอททริบิวต์อื่นเป็นเซตย่อย มาร่วมกันเพื่อให้เกิดเป็นเอกลักษณ์ในตารางนั้นๆ

##### 3. คีย์หลัก

คีย์หลัก (Primary Key) คือ คีย์คู่แข่งที่ถูกเลือก เพื่อกำหนดให้ข้อมูลในตารางนั้นมีความเป็นเอกลักษณ์ แยกให้แถวแต่ละแถวมีข้อมูลไม่ซ้ำกัน คำว่าแถว ในหนังสือบางเล่มเรียกว่าทูเปิล หรือ ระเบียบ โดยคุณสมบัติของคีย์หลักจะต้องไม่เป็นค่าว่าง ในแต่ละแอททริบิวต์อาจจะมีคีย์คู่แข่งที่ไม่ได้รับการเลือกให้เป็นคีย์หลัก คีย์นั้นจะเรียกว่าคีย์สำรอง (Alternate Key) ในการเลือกคีย์คู่แข่งเพื่อเป็นคีย์หลักอาจจะเลือกเอาแอททริบิวต์มากกว่าหนึ่งแอททริบิวต์ผสมกันได้เรียกว่า คีย์ผสม (Compound Key)

##### 4. คีย์นอก

คีย์นอก (Foreign Key) หมายถึง คีย์หลักของตารางแม่ เมื่อนำมาใช้ในการเชื่อมโยงในอีกตารางหนึ่ง (ตารางลูก) เพื่อให้ตารางสองตารางเชื่อมโยงข้อมูลถึงกัน

#### 2.2.5 คลังข้อมูล (Data Warehouse)

คลังข้อมูล หรือเรียกว่า ดาต้าแวร์เฮาส์ (Data Warehouse) คือที่เก็บรวบรวมข้อมูลจากหลายแหล่ง และหลากหลายชนิดเข้าด้วยกัน เข้าเป็นฐานข้อมูลขนาดใหญ่ โดยข้อมูลในคลังข้อมูลอาจได้มาจากฐานข้อมูลของระบบปฏิบัติการ และฐานข้อมูลจากแหล่งภายนอก ซึ่งข้อมูลเหล่านี้จะได้รับการเลือก การกลั่นกรอง การปรับแก้ไข และทำให้เป็นมาตรฐานเดียวกัน เพื่อช่วยให้ผู้ใช้หรือผู้ที่เกี่ยวข้องสามารถเข้าถึงข้อมูลได้อย่างสะดวกรวดเร็ว และมีข้อมูลที่พร้อมจะนำมาประมวลผลสำหรับกาสนับสนุนด้านการบริหารตัดสินใจ

### 2.2.5.1 ลักษณะของคลังข้อมูล

#### 1. การแบ่งโครงสร้างตามเนื้อหา (Subject Oriented)

คลังข้อมูลถูกออกแบบมาให้มีโครงสร้างตามเนื้อหาหลัก เช่น ตามรายการสินค้า ตามพื้นที่ เป็นต้น โดยจะจัดเก็บข้อมูลที่สนใจและเป็นประโยชน์ต่อการนำมาประมวลเพื่อสนับสนุนการตัดสินใจ

#### 2. การรวมเป็นหนึ่งเดียว (Integration)

ข้อมูลที่รวบรวมมาจากหลากหลายฐานข้อมูลจะถูกทำให้อยู่ในรูปแบบข้อมูลมาตรฐานเดียวกัน

#### 3. ความสัมพันธ์กับเวลา (Time Variant)

ข้อมูลที่จัดเก็บในคลังข้อมูลต้องกำหนดช่วงเวลาเอาไว้ เช่น อาจจัดเก็บข้อมูลย้อนหลัง 3 ปี หรือ 5 ปี เนื่องจากในการตัดสินใจจำเป็นต้องใช้ข้อมูลในการเปรียบเทียบในแต่ละช่วงเวลา หรือต้องใช้ข้อมูลในอดีตสำหรับคาดการณ์ในอนาคต

#### 4. ความเสถียรของข้อมูล (Nonvolatile)

ข้อมูลที่ถูกบันทึกไว้ในคลังข้อมูลจะไม่มีเปลี่ยนแปลงค่าหรือปรับปรุงข้อมูลเดิมที่มีอยู่แล้ว เนื่องจากเป็นข้อมูลที่ไม่มีการทวงถามทางธุรกิจมากระทบอีก

#### 5. ดาต้ามาร์ท (Data Mart)

คลังข้อมูลขนาดเล็กที่มีลักษณะเฉพาะเจาะจงมากกว่าสำหรับใช้ในองค์กรธุรกิจ ดาต้ามาร์ทมีขนาดของข้อมูลขนาดเล็กที่มีลักษณะเฉพาะเจาะจงมากกว่าสำหรับใช้ในองค์กรธุรกิจ ประโยชน์ที่เด่นชัดของดาต้ามาร์ท คือ ค่าใช้จ่ายที่ต่ำกว่า การจัดทำคลังข้อมูลก็ใช้เวลาที่สั้นกว่า และการนำไปประยุกต์ใช้ในการพัฒนาระบบสนับสนุนการตัดสินใจภายในหน่วยงานก็สะดวกกว่าการใช้คลังข้อมูลกลางขององค์กร

#### 6. ธุรกิจอัจฉริยะ (Business Intelligence)

เมื่อข้อมูลขององค์กรถูกจัดเก็บในคลังข้อมูล หรือดาต้ามาร์ทแล้ว ผู้จัดการ นักวิเคราะห์ และพนักงานสามารถที่จะเรียกใช้ข้อมูลเหล่านั้นตามสิทธิที่ตนได้รับ ผู้ใช้สามารถที่จะนำข้อมูลเหล่านั้นมาดำเนินการต่าง ๆ เพื่อใช้สนับสนุนการตัดสินใจ เช่น การจัดทำประวัติของลูกค้า การประเมินถึงสภาพของตลาด การจัดกลุ่มของตลาด การจัดลำดับทางด้านเครดิต การเพิ่มความสามารถในการทำกำไร และการจัดการความเคลื่อนไหวของสินค้าคงคลัง

#### 2.2.5.2 การวิเคราะห์ข้อมูลหลายมิติและการค้นหาความรู้ในคลังข้อมูล

เครื่องมือวิเคราะห์ข้อมูลและค้นหาความรู้ในคลังข้อมูลที่จะกล่าวถึงมี 2 ประเภท คือ การประมวลผลเชิงวิเคราะห์แบบออนไลน์ หรือ โอลาป (Online Analytical Processing : OLAP) และดาต้าไมนิ่ง (Data Mining)

การประมวลผลข้อมูลในคลังข้อมูลมักจะเกี่ยวข้องกับกับข้อมูลปริมาณมาก ๆ และข้อมูลหลายมิติ เช่น ข้อมูลยอดขายตามมิติของสินค้า ตามภูมิภาค และตามเวลา ซึ่ง OLAP เป็นเครื่องมือที่มีความสามารถในการค้นหาและวิเคราะห์ข้อมูลจากคลังข้อมูลเพื่อตอบสนองความต้องการของผู้ใช้ได้อย่างรวดเร็ว โดยสามารถใช้ OLAP ในการวิเคราะห์ข้อมูลในลักษณะต่าง ๆ เช่น

1. การหมุนมิติ (Rotation) สามารถวิเคราะห์หรือดูข้อมูลได้หลายมิติหรือได้หลายมุมมอง เช่น หาความสัมพันธ์ของยอดขาย สินค้า และภูมิภาค เป็นต้น
2. การเลือกช่วงข้อมูล (Ranging) สามารถเลือกดูข้อมูลเฉพาะส่วนที่สนใจ (Slice and Dice) และนำมาวิเคราะห์ได้โดยไม่ต้องใช้ข้อมูลทั้งหมด
3. การเลือกระดับชั้นของข้อมูล (Hierarchy) สามารถจัดแบ่งข้อมูลเป็นลำดับชั้น ทำให้เรียกดูข้อมูลจากระดับบนแล้วเจาะลึกลงไปในระดับล่าง (Drill-Down) เพื่อนดูรายละเอียดได้ เช่น ดูยอดขายรวม แล้วจึงดูยอดขายแบ่งตามภูมิภาค หรืออาจดูข้อมูลในรายละเอียดก่อนแล้วจึงดูที่ระดับบน (Roll-Up) ได้เช่นกัน

#### 2.2.5.3 ดาต้าไมนิ่ง (Data Mining )

ดาต้าไมนิ่ง เป็นเครื่องมือและเทคนิคในการสกัด (Extract) ข้อมูลและประมวลผลข้อมูลในเชิงวิเคราะห์ขั้นสูงจากฐานข้อมูลขนาดใหญ่ โดยสามารถค้นหารูปแบบ แนวโน้ม พฤติกรรมและความสัมพันธ์ที่ซ่อนอยู่ภายในข้อมูลเพื่อนำมาให้ความรู้ใหม่หรือคำตอบในลักษณะความสัมพันธ์ที่เกี่ยวข้อง (Association) ลำดับของข้อมูล (Sequence) การหากฎเกณฑ์ในการจัดกลุ่ม (Classification) การจัดกลุ่มของความสัมพันธ์ (Cluster) และการพยากรณ์ (Forecasting)

#### 2.2.5.4 ความแตกต่างระหว่างคลังข้อมูลกับฐานข้อมูลปฏิบัติการ

ฐานข้อมูลปฏิบัติการ (Operational Database) เป็นระบบที่นำมาช่วยในการรวบรวมและจัดการเก็บข้อมูลรายการจากการดำเนินงานประจำวัน ซึ่งข้อมูลเหล่านี้ในวันจะเพิ่มปริมาณมากขึ้น จึงจำเป็นต้องจัดเก็บข้อมูลในอดีตลงในสื่อบันทึกข้อมูลที่แยกเก็บต่างหาก เช่น จัดเก็บในเทป หรือ ดิสก์ที่มีความจุสูง เนื่องจากการประมวลผลข้อมูลที่มีปริมาณมากจะใช้เวลาานซึ่งส่งผลกระทบต่อการทำงานประจำวันได้

ประกอบกับข้อมูลที่จัดเก็บในฐานข้อมูลปฏิบัติการมักจะเก็บตามฟังก์ชันการทำงานในทางธุรกิจ มุ่งเน้นที่การลดความซ้ำซ้อนของข้อมูล

คลังข้อมูลเป็นแหล่งเก็บข้อมูลรวมขององค์กร ที่ได้รับการออกแบบมาให้มีความเรียบง่ายต่อการค้นหาและเรียกใช้งานได้อย่างรวดเร็ว สามารถรองรับการวิเคราะห์ข้อมูลที่มีปริมาณมาก โดยข้อมูลมีการจัดเก็บในลักษณะที่รวบรวม เป็นระเบียบเนื้อหา และแปรผันตามเวลา ซึ่งข้อมูลในคลังข้อมูลสามารถนำเข้าไปเป็นข้อมูลพื้นฐานให้กับระบบงานเพื่อการบริหารอื่นๆ และเหมาะกับการประมวลผลเชิงวิเคราะห์ที่เป็นประโยชน์ต่อการประกอบการตัดสินใจของผู้บริหาร

## 2.2.6 การศึกษาการคำนวณพื้นฐานอ้างอิงจากสิ่งที่ผู้ประกอบการต้องการ [4], [5]

### 2.2.6.1 การบริหารจัดการสินค้าคงคลัง

รับผิดชอบในเรื่องของการบริหารระดับสินค้าคงคลังทั้งหมด ครอบคลุมตั้งแต่การพัฒนาระบบการเติมเต็มสินค้าทั้งที่ศูนย์กระจายสินค้าและสาขา โดยการตรวจสอบและกำหนดค่าจุดสั่งซื้อเพิ่ม (Min) และปริมาณการสั่งซื้อ โดยใช้ดุลยพินิจประสบการณ์ประกอบกับข้อมูลในอดีตเป็นแนวทางในการพิจารณา การวิเคราะห์ระดับการหมุนเวียนของสินค้าสินค้าแต่ละกลุ่ม (Inventory Turn Over) ให้เป็นไปตามเกณฑ์ที่กำหนด ตลอดจนการกำหนดนโยบายสินค้าคงคลังอื่นๆ เช่น การแบ่งหมวดสินค้า ว่าสินค้าประเภทใดจัดอยู่ในหมวดของสินค้าที่เคลื่อนไหวเร็ว ปานกลาง ช้าหรือจัดอยู่ในหมวดสินค้าที่ไม่มีการเคลื่อนไหว การพิจารณาสินค้าใหม่ การจัดประเภทสินค้าโปรโมชั่น รวมทั้งกำหนดแผนปฏิบัติในทุกขั้นตอน

### 2.2.6.2 การจำแนกหมวดสินค้า

บริษัทกรณีศึกษา มีการจำแนกหมวดสินค้า ตามปริมาณความต้องการสินค้า แต่ละประเภท ตามโครงสร้างสินค้า ดังนี้

1. หมวดสินค้าที่มีการเคลื่อนไหวเร็ว (Fast Moving)
2. หมวดสินค้าที่มีการเคลื่อนไหวปานกลาง (Medium Moving)
3. หมวดสินค้าที่มีการเคลื่อนไหวช้า (Low Moving)
4. สินค้าที่ไม่มีการเคลื่อนไหว (Dead Stock)

### 2.2.6.3 การควบคุมวัสดุคงคลัง (Inventory Control)

การควบคุมวัสดุคงคลังเป็นสิ่งสำคัญอย่างหนึ่งที่จะต้องนำมาพิจารณาในการดำเนินธุรกิจ ทั้งนี้เพราะการมีวัสดุคงคลังนั้นจำเป็นต้องใช้เงินทุน ซึ่งมีมูลค่าสูงในกลุ่มทรัพย์สินหมุนเวียน ดังนั้นจึงต้องเฝ้าติดตามดูระดับวัสดุคงคลังอยู่เสมอ และจัดทำให้มีจำนวนที่เพียงพอต่อการผลิต หรือเพื่อจัดจำหน่ายให้ลูกค้า อย่างไรก็ตามการมีวัสดุคงคลังในระดับต่ำนั้นย่อมเป็นที่ต้องการของผู้ประกอบการ เพราะมีผลต่อค่าประกันเงินทุน การจัดเก็บ ตลอดจนการจัดการด้านวัสดุคงคลัง ในทางตรงกันข้าม ถ้ามีวัสดุคงคลังในระดับสูง จะมีผลทำให้ผลตอบแทนจากการลงทุนลดน้อยลง

ในกรณีที่วัสดุคงคลังไว้ต่ำเกินไป จะมีผลทำให้ประสิทธิภาพในการดำเนินการผลิตลดลง สินค้ามีไม่พอขาย ลูกค้าผิดหวัง และวัตถุดิบมีต้นทุนสูงขึ้น ดังนั้น จะเห็นได้ว่าการมีวัสดุคงคลังต่ำหรือสูงเกินไปก็ย่อมจะไม่ก่อให้เกิดผลดีต่อการดำเนินงานของธุรกิจ เพื่อป้องกันไม่ให้เกิดปัญหาด้านวัสดุคงคลังดังกล่าวจึงควรมีการจัดการวัสดุคงคลัง เช่น การหาจำนวนการสั่งซื้อสินค้าที่เหมาะสมและประหยัด การหาจุดสั่งซื้อ และการหาสต็อกเพื่อความปลอดภัย (Safety Stock)

### 2.2.6.4 การคงคลังสินค้าสำเร็จรูป (Finish Goods)

ถ้าเราสามารถกำหนดจำนวนวัตถุดิบที่ต้องการใช้ได้อย่างแน่นอนในแต่ละช่วงเวลาแล้ว นั่นก็หมายความว่า เราสามารถหาปริมาณของสินค้าที่จะผลิตที่พอดีกับความต้องการที่เราคาดหวังไว้ แต่ว่าความต้องการสินค้าของลูกค้า นั้นมักจะไม่ค่อยมีความแน่นอน คือ มีการเปลี่ยนแปลงตลอดเวลา ดังนั้น การมีสินค้าสำเร็จรูปคงคลังไว้ก็ย่อมจะก่อให้เกิดประโยชน์ในแง่ต่างๆ เช่น

1. เพื่อป้องกันการขาดแคลนสินค้า ในกรณีที่ความต้องการมีมากกว่าค่าที่ได้พยากรณ์หรือปริมาณที่จัดหามาได้มีน้อยกว่าจำนวนที่คาดหวังไว้ในแต่ละช่วงเวลา ดังนั้น จึงจำเป็นต้องมีการสต็อกสินค้าสำเร็จรูปไว้จำนวนหนึ่งเพื่อความปลอดภัย ซึ่งโดยปกติแล้วจะเสียค่าใช้จ่ายน้อยกว่าการสั่งผลิตเพิ่มเติม (Back Order) ทั้งยังจะไม่เป็นการเสี่ยงต่อการสูญเสียความเชื่อถือ หรือเสียลูกค้าในกรณีที่สินค้าเกิดการขาดตลาด

2. เพื่อช่วยให้การผลิตสินค้าสามารถดำเนินต่อไปได้อย่างสม่ำเสมอและเป็นไปอย่างต่อเนื่อง

### 2.2.6.5 การตัดสินใจขั้นพื้นฐานวัสดุคงคลัง (Basic Inventory Decisions)

โดยอาศัยหลักการจัดการด้านนโยบายวัสดุคงคลัง ที่พยายามจะลดค่าใช้จ่ายในการดำเนินงาน (Operation Cost) ของธุรกิจให้ต่ำสุด ซึ่งจะต้องพิจารณาหลักการ 2 ประการ

1. จำนวนที่จะต้องสั่งซื้อในแต่ละครั้ง
2. จะพิจารณาว่าเมื่อใดจึงจะสั่งซื้อวัสดุจำนวนนี้

การพิจารณาหาแนวทางการตัดสินใจเป็นไปได้ว่า การสั่งซื้อเป็นจำนวนมากเพื่อลดค่าใช้จ่ายในการสั่งซื้อให้ต่ำสุด หรือสั่งซื้อครั้งละน้อยๆ เพื่อลดค่าใช้จ่ายในการเก็บวัสดุคงคลังให้ต่ำสุด ทางที่จะทำให้ได้ประโยชน์สูงสุดนั้น จะไม่ได้เกิดจากการเลือกทางหนึ่งทางใด แต่จะต้องเลือกระหว่าง 2 ทาง

#### 1. ค่าใช้จ่ายวัสดุคงคลัง (Inventory Cost)

ค่าใช้จ่ายของธุรกิจที่เกิดจากการคงคลัง สามารถสรุปได้เป็น 3 ประเภทใหญ่ คือ ค่าใช้จ่ายในการสั่งซื้อ (Ordering Cost) และค่าใช้จ่ายในการจัดเก็บ (Carrying Cost) ซึ่งจะนำมาเทียบให้เท่ากันสำหรับตัวแบบการคงคลังของเรา ส่วนค่าใช้จ่ายประเภทที่ 3 เกิดจากการสูญเสียเนื่องจากไม่มีสินค้าเก็บไว้ในสต็อก ซึ่งจำเป็นจะต้องมีการสั่งเพิ่มเติม (Back Order)

#### 2. นิยามของวัสดุคงคลังโดยเฉลี่ย

ก่อนที่จะปรับปรุงตัวแบบของขนาดวัสดุคงคลัง จำเป็นต้องมีสมมติฐานก่อน โดยกำหนดให้การสั่งซื้อ (Purchase) วัสดุสำหรับคงคลังจะเป็นเพียงชนิดเดียว (Single Item) โดยมีอุปสงค์ (Demand) ในอัตราคงที่และผู้ทำการตัดสินใจรู้ค่าก่อนล่วงหน้า เช่นเดียวกับเวลานำ (Lead Time) ซึ่งเป็นเวลาระหว่างที่ทำการสั่งจนกระทั่งรับวัสดุเข้าคลัง ถึงแม้ว่าข้อสมมติฐานดังกล่าวนี้ยากที่จะเป็นไปได้สำหรับปัญหาคลังในธุรกิจจริงๆ แต่เราก็สามารถจะพัฒนาตัวแบบอย่างธรรมดานี้ได้ โดยใส่ค่าตัวแปรต่างๆ (Factors) ที่เป็นจริงลงไปปริมาณของการคงคลังที่เวลาใดๆ ภายใต้สมมติฐานที่กล่าวมาแล้ว

ถ้าให้  $Q$  เป็นปริมาณของการสั่งซื้อ (Order Size) และจะมีจำนวนคงคลังเท่ากับ  $Q$  เมื่อได้รับวัสดุแล้ว วัสดุนั้นก็จะค่อยๆ ถูกนำมาใช้และค่อยๆ หมดไปจนเป็นศูนย์ และจุดนี้เองวัสดุที่ได้สั่งซื้อไว้แล้วจะถูกนำมาเติมเต็ม (Replenish) อีก จึงสังเกตได้ว่าจำนวนวัสดุคงคลังโดยเฉลี่ย ( $Q/2$ ) จะเท่ากับครึ่งหนึ่งของจำนวนที่สั่งซื้อแต่ละครั้ง (Lot Size) การสั่งซื้อแต่ละครั้งจนได้รับวัสดุเข้าคลัง จะมีเวลาและค่าใช้จ่ายที่แน่นอน ดังนั้นจึงไม่ก่อให้เกิดปัญหาการขาดสต็อกแต่อย่างใด

#### 2.2.6.6 ข้อควรพิจารณาในการควบคุมวัสดุคงคลัง

ระบบควบคุมวัสดุคงคลัง ซึ่งเราให้ความสนใจถึงหลักเกณฑ์ในการตัดสินใจที่เกี่ยวกับการสั่ง เช่น เมื่อไรจึงจะสั่ง และจะสั่งเป็นจำนวนเท่าไร ในทางปฏิบัติแล้วกฎเกณฑ์ต่างๆ เหล่านี้จะอยู่ในเนื้อหาของระบบควบคุมวัสดุคงคลัง นอกเหนือจากการคำนวณหาค่าต่างๆ เพื่อนำมาใช้ทำการตัดสินใจแล้ว ระบบดังกล่าวควรจะต้องระบุถึงแนวทางการจัดบันทึก การดำเนินงานวัสดุคงคลัง และติดตามผล เพื่อหาประสิทธิภาพในการจัดการคงคลังด้วย

ระบบควบคุมวัสดุคงคลัง สามารถจะกระทำได้ด้วยวิธีการธรรมดา เช่น การทำด้วยมือหรือใช้เครื่องคอมพิวเตอร์ หรือใช้ทั้งมือและคอมพิวเตอร์ร่วมกัน แต่ในปัจจุบันการใช้คอมพิวเตอร์เป็นที่นิยมกันโดยทั่วไป ยกเว้นในกรณีที่สินค้ามีจำนวนน้อยรายการและราคาแพง จึงไม่เหมาะสมที่จะนำคอมพิวเตอร์มาใช้ เพราะผลประโยชน์ที่ได้รับจะไม่คุ้มกับรายจ่ายที่ต้องเสียไป

ไม่ว่าระบบควบคุมจะเป็นแบบไหน จะใช้เครื่องคอมพิวเตอร์หรือไม่ก็ตาม แต่หน้าที่ที่จะต้องปฏิบัติมีดังต่อไปนี้

#### 1. การจักระบบวัสดุคงคลัง

ระบบที่ใช้สำหรับการคงคลังทุกระบบ จำเป็นต้องมีวิธีการจัดเก็บข้อมูล ซึ่งจะเป็นประโยชน์ต่อระบบบัญชีและงานด้านการจัดการคงคลัง บางครั้งมีความจำเป็นต้องจัดบันทึกรายรับและรายจ่ายทุกครั้งที่เกิดขึ้น แต่ในบางกรณีการบันทึกเป็นช่วงๆ ก็อาจเพียงพอแล้ว

#### 2. กฎเกณฑ์การตัดสินใจสำหรับวัสดุคงคลัง

กฎเกณฑ์การตัดสินใจที่เกี่ยวข้องกับระบบการคงคลังคือ การหาว่าเมื่อไหร่ควรจะสั่งและสั่งเป็นจำนวนเท่าไร

#### 3. รายงานในเรื่องข้อยกเว้น

เมื่อมีการตัดสินใจที่จะใช้กฎเกณฑ์อย่างไรกับระบบ การรายงานข้อยกเว้นต่างๆ ก็ควรจะเสนอให้กับฝ่ายบริหารด้วย ข้อยกเว้นเหล่านี้อาจจะรวมถึงสถานการณ์ที่การพยากรณ์ไม่สามารถติดตามค่าอุปสงค์ได้อย่างใกล้เคียง จำนวนขาดสต็อกและอื่นๆ จุดประสงค์ของการรายงานสำหรับข้อยกเว้น ก็เพื่อให้ฝ่ายบริการเตรียมพร้อมในการเปลี่ยนแปลงสมมติฐานในทางปฏิบัติถ้าระบบใดไม่มีการประสานกับรายงานข้อยกเว้นแล้ว ก็อาจจะทำให้ระบบนั้นนอกนอกการควบคุมซึ่งจะเกิดผลเสียในเรื่องของการประหยัด

#### 4. การพยากรณ์

การตัดสินใจเกี่ยวกับวัสดุคงคลัง ควรจะขึ้นอยู่กับพยากรณ์ของอุปสงค์ การตัดสินใจกับวัสดุคงคลังไม่ควรจะขึ้นอยู่กับแผนขายหรือผู้จัดการฝ่ายควบคุมสต็อกแต่เพียงฝ่ายเดียวควรจะใช้เทคนิคด้านประมาณเข้ามาช่วยเสริมกับระบบที่มีอยู่ และในบางครั้งจำเป็นต้องปรับปรุงเทคนิคการพยากรณ์ด้านการประมาณให้เข้ากับเหตุการณ์ที่ไม่ปกติด้วย

#### 5. การรายงานต่อผู้บริหารระดับสูง

ระบบการควบคุมวัสดุคงคลัง รายละเอียดของรายงานจะแสดงถึงผลการวัดประสิทธิภาพการทำงานของคงคลังทั้งหมด ซึ่งจะเป็นการช่วยในการกำหนดนโยบายอย่างกว้างๆ ในรายงานนี้ควรจะรวมถึงระดับการให้บริการ (Service Level) ค่าใช้จ่ายในการดำเนินการคงคลัง และระดับการ

ลงทุน โดยการเปรียบเทียบกับช่วงเวลาอื่นๆ โดยทั่ว ๆ ไป ความเชื่อส่วนใหญ่จะมุ่งไปที่ประเด็นการวัดประสิทธิภาพ โดยดูจากอัตราส่วนของการหมุนเวียน (Turn Over Ratios) ซึ่งผลที่ได้จะเป็นส่วนหนึ่งของข้อมูลแต่ยังไม่เพียงพอที่จะนำมาใช้ในการกำหนดนโยบายการคลัง ในทางปฏิบัติยังมีระบบจำนวนมากที่ยังให้ข้อมูลไม่ดีพอกับฝ่ายบริหารระดับสูง

ดังนั้นจะเห็นได้ว่า ระบบการควบคุมวัสดุคงคลังที่ดีจะไม่เพียงพอแต่ให้รายละเอียดในการบันทึกรายการวัสดุต่างๆ เท่านั้น แต่ยังใช้สำหรับการควบคุมระดับการคงคลัง นอกจากนี้ยังต้องเป็นประโยชน์สำหรับการจัดการเพื่อการตัดสินใจอีกด้วย

#### 2.2.6.7 ระบบช่วงเวลาการสั่งซื้อคงที่ (Periodic Review System or Fixed Time Period)

ระบบช่วงเวลาการสั่งซื้อคงที่ วัสดุคงคลังจะมีการตรวจนับตามช่วงเวลาที่กำหนด ยกตัวอย่างเช่น ทุกสัปดาห์หรือทุกเดือน หลังจากที่ได้มีการตรวจนับแล้ว ระยะเวลาในการสั่งซื้อคงที่ แต่ปริมาณในการสั่งแต่ละครั้งไม่คงที่ ส่วนใหญ่แล้วผู้ค้าปลีกทั่วไปจะใช้ระบบนี้ในการพิจารณาระดับคลังสินค้าโดยสินค้าจะเป็นสินค้าที่มีราคาไม่สูงมากนัก แต่ถ้าหากเป็นสินค้าที่มีความสำคัญมาก ก็จำเป็นที่จะต้องมีการสั่งสินค้าเป็นสต็อกเพื่อความปลอดภัยไว้ด้วย

#### 2.2.6.8 จุดสั่งซื้อกรณีความต้องการผันแปร (Reorder Point with Variable Demand)

ในการคำนวณจุดสั่งซื้อกับสต็อกเพื่อความปลอดภัยที่จะต้องอยู่ในระดับบริการ จะต้องกำหนดให้ความต้องการในแต่ละวันในช่วงเวลานำของการสั่งซื้อไม่แน่นอน เป็นอิสระต่อกัน และสามารถอธิบายได้ด้วยการกระจายแบบปกติ ค่าเฉลี่ยของความต้องการในช่วงเวลานำคือ ผลรวมของความต้องการเฉลี่ยในแต่ละวันในช่วงเวลานำ หรือก็คือผลคูณของค่าเฉลี่ยของความต้องการในแต่ละวันกับช่วงเวลานำ ( $\bar{d}L$ ) ส่วนความแปรปรวนของการกระจายก็คือ ผลรวมของความแปรปรวนในแต่ละวันสำหรับในช่วงเวลานำ ดังนั้นเราสามารถหาค่าของจุดสั่งซื้อในกรณีความต้องการผันแปรเพื่อให้อยู่ในระดับบริการที่ต้องการได้คือ

$$R = \bar{d}L + z\sigma_d\sqrt{L} \quad (2.1)$$

เมื่อ	$\bar{d}$	=	อัตราเฉลี่ยของความต้องการสินค้า
	L	=	ช่วงเวลานำ
	$\sigma_d$	=	ค่าเบี่ยงเบนมาตรฐานของความต้องการต่อวัน
	Z	=	จำนวนของค่าเบี่ยงเบนมาตรฐานที่สอดคล้องกับระดับความน่าจะเป็นของการให้บริการ
	$z\sigma_d\sqrt{L}$	=	สต็อกเพื่อความปลอดภัย

### 2.2.6.9 ปริมาณสั่งซื้อกรณีต้องการผันแปร (Order Quantity with Variable Demand)

ถ้าอัตราความต้องการและช่วงเวลานำของการสั่งซื้อคงที่ ตัวแบบของระบบการสั่งซื้อตามช่วงเวลาจะกำหนดให้ปริมาณการสั่งซื้อคงที่ด้วย ซึ่งก็จะมีค่าเท่ากับปริมาณการสั่งอย่างประหยัด แต่ถ้าหากความต้องการผันแปร ปริมาณที่จะสั่งก็จะแตกต่างกันออกไป

ปริมาณการสั่งสำหรับตัวแบบช่วงเวลาการสั่งซื้อคงที่ที่มีความต้องการสินค้าแต่ละวันที่ผันแปรไปในลักษณะการกระจายแบบปกติพิจารณาได้จาก

$$Q = \bar{d}(t_b + L) + z\sigma_d\sqrt{t_b + L - I} \quad (2.2)$$

เมื่อ	$\bar{d}$	=	อัตราเฉลี่ยของความต้องการสินค้า
	$t_b$	=	ช่วงเวลาคงที่ระหว่างการสั่งซื้อ
	L	=	ช่วงเวลานำของการสั่งซื้อ
	$\sigma_d$	=	ค่าเบี่ยงเบนมาตรฐานของความต้องการสินค้า
	$z\sigma_d\sqrt{t_b + L - I}$	=	สต็อกเพื่อความปลอดภัย
	I	=	ระดับของพัสดุกคงคลัง

#### 2.2.6.10 อัตราการหมุนเวียนของสินค้าคงคลัง (Inventory Turnover)

อัตราการหมุนเวียนของสินค้า (Inventory Turnover) สามารถใช้เป็นดัชนีการผลิขององค์กร สามารถตอบสนองความต้องการของตลาดได้หรือไม่ และการบริหารสินค้าคงคลังได้ดีแค่ไหนเมื่อเทียบกับการขาย ในทางธุรกิจการเปลี่ยนวัตถุดิบและสินค้าเป็นเงินให้เร็วที่สุด มีส่วนสำคัญในทางธุรกิจทำให้สามารถบริหารงบการเงินขององค์กรเป็นไปอย่างมีประสิทธิภาพสูงสุด

$$\text{อัตราการหมุนเวียนของวัสดุคงคลัง} = \frac{\text{ต้นทุนขาย}}{\text{ค่าเฉลี่ยของวัสดุคงคลัง}} \quad (2.3)$$

อัตราการหมุนเวียนของสินค้าจึงพิจารณาถึงมูลค่าสินค้าที่มีอยู่จะต้องมีการหมุนเวียนอย่างไรเพื่อให้เกิดประสิทธิภาพสูงสุดในรอบระยะเวลาเฉลี่ย 12 เดือน (ตามปกติ) หรือตามรอบระยะเวลาที่ต้องการคำนวณ เพื่อป้องกัน ความผันผวนในปริมาณของสินค้าคงเหลือ

### 2.3 NETBEANS IDE [6], [7]

โปรแกรม NETBEANS IDE คือ เครื่องมือสำหรับโปรแกรมเมอร์ที่จะใช้พัฒนา Application ด้วย ภาษา Java NetBeans นั้นเป็นโปรแกรมประเภท Open Source Software โดยผู้ใช้งานไม่จำเป็นต้องเสียเงิน เพื่อซื้อมาใช้งาน และยังเปิดเผย Source code ให้ผู้สนใจและนักพัฒนานำไปดัดแปลง แก้ไข ตามกฎของ Open Source โดยในปัจจุบัน NetBeans ได้รับการพัฒนาให้มีความสามารถสูงยิ่งขึ้นเรื่อยๆ จนถึงเวอร์ชันล่าสุด คือ นอกจากจะใช้ในการพัฒนา Application ด้วยภาษาจาวาแล้ว ยังสามารถพัฒนาอื่นๆได้อีกหลากหลายโดยการติดตั้งโปรแกรมเสริม (Add-On) ได้จากเว็บไซต์หรือผ่านตัวอัปเดตเซนเตอร์ (Update Center) ของ NetBeans เช่น ภาษาซี/ซีพลัสพลัส (C/C++), Ruby, UML, SOA, เว็บแอปพลิเคชัน, Java EE, Mobility (Java ME), Java FX, Java Script, PHP เป็นต้น ในเวอร์ชัน 6.0 เป็นต้นไปมีการรวมโปรแกรมเสริมต่างๆที่สำคัญเข้าในตัวติดตั้งของ NetBeans โดยสามารถเลือกติดตั้งได้ภายหลัง

### 2.3.1 ส่วนประกอบของ NetBeans

หน้าจอหลักของ NetBeans จะแบ่งออกเป็น 4 ส่วนใหญ่ๆ

- 1.เป็นส่วนของแถบเมนู และ Tool ต่างๆ
- 2.เป็นส่วนที่แสดงและจัดการ Project ต่างๆ เหมือนเป็นการ Browse ดู Project หรือ File ต่างๆ
- 3.เป็นส่วนที่ใช้ในการเขียน Code
- 4.เป็นส่วนที่ใช้ในการแสดงผลการทำงานต่าง เมื่อทำการ Run โปรแกรม และยังมีส่วนของการ Debug โปรแกรมด้วย

### 2.3.2 ภาษา JAVA

ภาษา Java มีรูปแบบ Syntax คล้าย ๆ กับภาษา C/C++ ในการเขียนภาษาจะเป็นภาษาโปรแกรมเชิงวัตถุที่ค่อนข้างจะสมบูรณ์แบบ เพราะฉะนั้นการเขียนด้วย Syntax ของภาษา Java นั้น เราจะเขียนในมุมมองของ OOP เป็นหลักสำหรับการพัฒนาภาษา Java ในการพัฒนาภาษา เหตุผลหลักที่ทำให้สนใจที่จะศึกษาภาษา Java ก็คือ สามารถใช้โปรแกรมสะดวก โดยรวบรวมข้อมูลไว้ที่ศูนย์กลาง และใช้จัดการสินค้าคลังได้ทุกที่

### 2.3.3 MySQL

MySQL คือ โปรแกรมระบบจัดการฐานข้อมูล ที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่นอย่างบูรณาการ เพื่อให้ได้ระบบงานที่รองรับ ความต้องการของผู้ใช้ เช่นทำงานร่วมกับเครื่องบริการเว็บ (Web Server) โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลโอเพนซอร์ซ (Open Source) ที่ถูกนำไปใช้งานมากที่สุด ซึ่งมีรายละเอียดดังนี้

1. ข้อกำหนดในการเชื่อมต่อ MySQL ระหว่าง Server ทั้ง 2 เครื่องสามารถติดต่อสื่อสารกันได้ผ่านระบบ Lan หรือ Internet MySQL Database เครื่องที่เป็น Server จะต้อง Allow Port 3306 และ Network ของ Client ก็ 3306 เช่นเดียวกันกำหนด Host สำหรับ User เป็น IP ของเครื่อง Client หรือจะใช้เปอร์เซ็นต์สำหรับเครื่องใดๆก็ได้ (ปกติเป็น Local Host)

## 2. ความสามารถและการทำงานของโปรแกรม MySQL มีดังต่อไปนี้

2.1 MySQL ถือเป็นระบบจัดการฐานข้อมูล (Database Management System (DBMS) ฐานข้อมูลมีลักษณะเป็นโครงสร้างของการเก็บรวบรวมข้อมูล การที่จะเพิ่มเติม เข้าถึงหรือประมวลผล ข้อมูลที่เก็บในฐานข้อมูลจำเป็นจะต้องอาศัยระบบจัดการ ฐานข้อมูล ซึ่งจะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูลสำหรับการใช้งานเฉพาะ และรองรับการทำงานของแอปพลิเคชันอื่นๆ ที่ต้องการใช้งานข้อมูลในฐานข้อมูล เพื่อให้ได้รับความสะดวกในการจัดการกับข้อมูลจำนวนมาก MySQL ทำหน้าที่เป็นทั้งตัวฐานข้อมูล และระบบจัดการฐานข้อมูล

2.2 MySQL เป็นระบบจัดการฐานข้อมูลแบบ Relational ฐานข้อมูลแบบ Relational จะทำการเก็บข้อมูลทั้งหมดในรูปแบบของตารางแทนการเก็บข้อมูลทั้งหมดลงในไฟล์ เพียงไฟล์เดียว ทำให้ทำงานได้รวดเร็วและมีความยืดหยุ่น นอกจากนั้นแต่ละตารางที่เก็บข้อมูลสามารถเชื่อมโยงเข้าหากันทำให้สามารถรวมหรือจัดกลุ่มข้อมูลได้ตามต้องการ

2.3 MySQL ใช้งานแบบ Open Source ซึ่งสามารถใช้งาน และปรับแต่งการทำงานได้ตามต้องการ โดยเฉพาะความสามารถในการทำงานรวดเร็ว น่าเชื่อถือและใช้งานได้ง่าย นอกจากนั้น MySQL ถูกออกแบบและพัฒนาขึ้นมาเพื่อทำหน้าที่เป็นเครื่องให้บริการรองรับการจัดการกับฐานข้อมูลขนาดใหญ่ ซึ่งการพัฒนา ยังคงดำเนินอยู่อย่างต่อเนื่อง ส่งผลให้มีฟังก์ชันการทำงานใหม่ๆ ที่อำนวยความสะดวกแก่ผู้ใช้งานเพิ่มขึ้นอยู่ตลอดเวลา รวมไปถึงการปรับปรุงด้านความต่อเนื่อง ความเร็วในการทำงาน และความปลอดภัย ทำให้ MySQL เหมาะสมต่อการนำไปใช้งานเพื่อเข้าถึงฐานข้อมูลบนเครือข่ายอินเทอร์เน็ต

## บทที่ 3

### การออกแบบและวิธีการดำเนินการ

การจัดทำวิทยานิพนธ์เรื่องการออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น เพื่อจัดการกับปัญหาด้านการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น และพัฒนาระบบสนับสนุนการตัดสินใจที่มีคุณภาพและประสิทธิภาพในการรองรับข้อมูลภายในคลังสินค้าที่มีการหมุนเวียนสินค้าตลอดเวลา ซึ่งการออกแบบระบบสนับสนุนการตัดสินใจนั้นถูกพัฒนานาในรูปแบบของเว็บแอปพลิเคชันผ่านทางโปรแกรม NetBeans IDE 8.1 ซึ่งมีขั้นตอนและวิธีการดำเนินงานดังนี้

1. ศึกษารูปแบบธุรกิจ และเก็บข้อมูลปัญหาจากที่เกิดขึ้นจากผู้ประกอบการ
2. ศึกษาทฤษฎีที่เกี่ยวข้องกับวิทยานิพนธ์ วิเคราะห์ปัญหา และเครื่องมือที่ใช้ในการแก้ไขปัญหา
3. จัดการโครงสร้างฐานข้อมูลด้วยรูปแบบ ER Diagram และจัดเรียงข้อมูลผ่าน Microsoft Excel
4. วางแผนการพัฒนาระบบผ่านรูปแบบของเว็บแอปพลิเคชันโดยโปรแกรม NetBeans IDE 8.1
5. ทดสอบระบบการทำงานของเว็บแอปพลิเคชัน และสรุปผลการทำงานของระบบ
6. จัดทำรูปเล่มปริญาวิทยานิพนธ์และตรวจสอบความถูกต้อง

#### 3.1 ศึกษารูปแบบธุรกิจ และเก็บข้อมูลปัญหาจากที่เกิดขึ้นจากผู้ประกอบการ

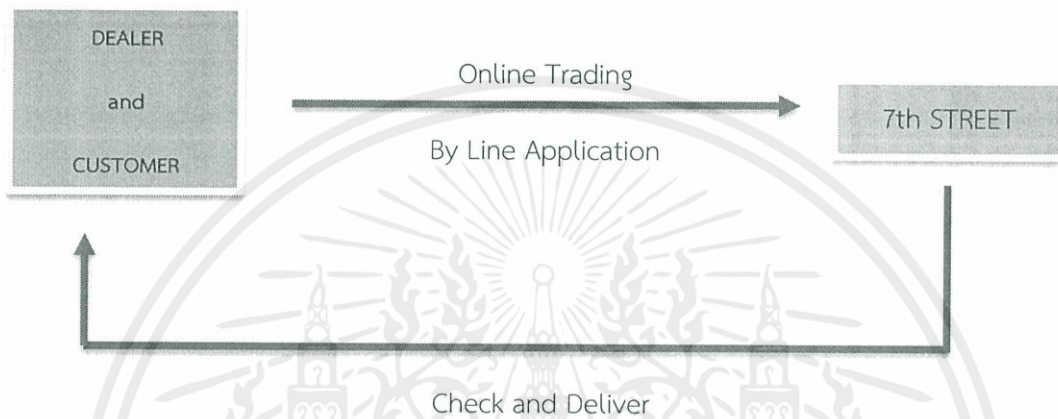
การศึกษารูปแบบธุรกิจที่เลือกมาใช้ในการศึกษาและพัฒนาระบบการจัดการสินค้าคงคลัง เป็นการศึกษาเกี่ยวกับรูปแบบของธุรกิจเสื้อผ้าแฟชั่น โดยรับการสนับสนุนด้านฐานข้อมูลจาก บริษัท 7<sup>TH</sup> Street เพื่อนำข้อมูลมาใช้ในการวิเคราะห์และจัดการฐานข้อมูลตัวอย่างในการสร้างระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลัง โดยเราทำการศึกษารูปแบบของธุรกิจโดยรวมตั้งแต่ภาพรวมขององค์กร รูปแบบของการซื้อขายสินค้า และขั้นตอนการจัดการภายในองค์กร เพื่อทำความเข้าใจรูปแบบของธุรกิจ

##### 3.1.1 ข้อมูลและรูปแบบของธุรกิจเสื้อผ้าแฟชั่น

7th Street เป็นบริษัทเสื้อผ้าแฟชั่น ที่มีรูปแบบการขายผ่านทางช่องทางออนไลน์และรูปแบบการขายแบบ dealer ซึ่งการขายแบบออนไลน์นั้นจะเป็นการขายสินค้าผ่านทางโซเชียลเน็ตเวิร์ค สั่งซื้อโดยตรงผ่าน แอปพลิเคชัน Line ขึ้นตรงกับบริษัท แล้วบริษัทก็จะทำการจัดส่งสินค้าไปแก่ลูกค้า แต่รูปแบบการขาย

แบบ dealer นั้นจะมีการจัดสรรให้แต่ละพื้นที่ที่มีการขายที่ไม่ทับซ้อนกัน เพื่อลดปัญหาการแข่งขันกันเองในแต่ละผู้ประกอบการ

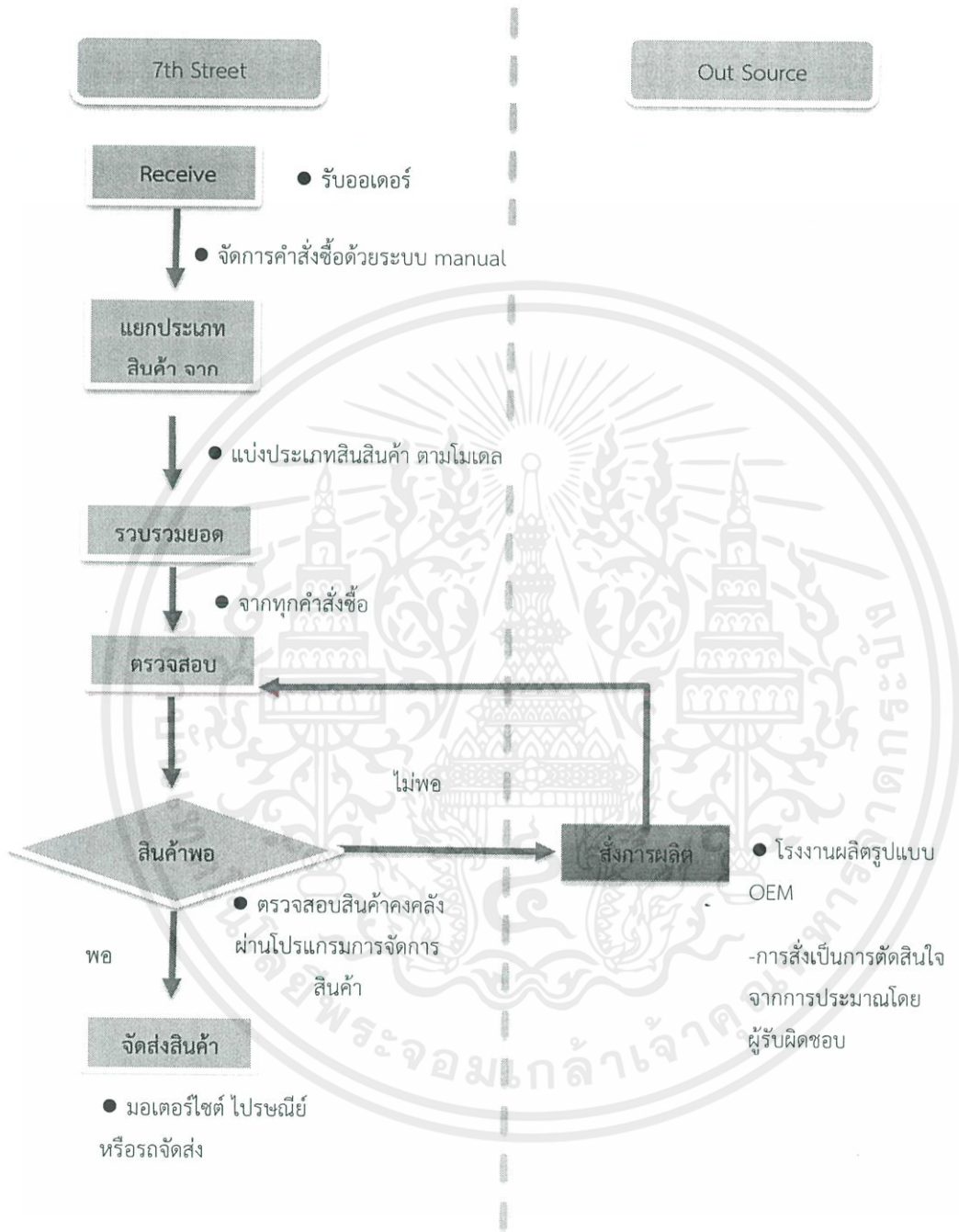
7th Street แบ่งบริษัทออกเป็นสองส่วน คือ ส่วนหน้าร้านหลักจะอยู่ที่ แพลตตินั่ม ส่วนในการจัดการ warehouse จะอยู่ที่กรุงเทพฯ โดยมีแผนผังธุรกิจโดยรวมดังรูปที่ 3.1



รูปที่ 3.1 แผนผังธุรกิจโดยรวม

### 3.1.2 รูปแบบการบริหารภายใน

เมื่อมีคำสั่งซื้อจากลูกค้าถึงบริษัท บริษัทก็จะทำการตรวจสอบสินค้าเบื้องต้นแล้วตอบกลับคำสั่งซื้อเพื่อแจ้งเวลาที่พร้อมจะทำการจัดส่งไปถึงมือลูกค้า หลังจากนั้นบริษัทก็จะทำการรวบรวมคำสั่งซื้อตามแยกตามรูปแบบของโมเดล เพื่อรวบรวมยอดการสั่งซื้อทั้งหมด จะทำให้ทราบจำนวนสุทธิของแต่ละโมเดลสินค้า เมื่อทราบจำนวนคำสั่งซื้อมาแล้ว จะไปทำการตรวจจำนวนสินค้าใน warehouse ว่ามีจำนวนเพียงพอหรือไม่ หากมีจำนวนเพียงพอก็จะทำการจัดส่งสินค้า หากสินค้ามีจำนวนไม่เพียงพอหรือเพียงพอแต่มีจำนวนคงเหลือเล็กน้อย ก็จะมีการสั่งการผลิตไปสู่อู่บริษัทที่ทำหน้าที่ผลิตสินค้าให้ โดยรูปแบบการสั่งการผลิตนั้น ใช้รูปแบบการ forecast ตามประสบการณ์ จากการคาดการณ์ ของผู้ประกอบการนั้นๆ เมื่อสินค้าที่สั่งผลิตกลับมาแล้วนั้น จะกลับเข้ามาอยู่ใน warehouse เพื่อรอการจัดส่งต่อไป โดยในปัจจุบัน ทางบริษัทมีตัวช่วยในการจัดการ warehouse คือ ใช้โปรแกรมทั่วไปในการบริหารจัดการสต็อก เป็นโปรแกรมในรูปแบบออนไลน์ ไม่สามารถเชื่อมข้อมูลถึงกันภายในองค์กรได้ โดยโปรแกรมมีความสามารถเพียงการตรวจสอบสถานภาพของสินค้า ว่ามีเข้ามา หรือ ออกไปเท่าใด และมีจำนวนคงคลังเหลือเท่าไร ซึ่งสามารถสรุปแผนผังได้ดังรูปที่ 3.2



รูปที่ 3.2 แผนผังธุรกิจโดยละเอียด

### 3.1.3 การรับทราบปัญหาและความต้องการจากผู้ประกอบการ

จากลักษณะรูปแบบของ Business Flow และข้อมูลการบริหารจัดการของบริษัท 7th Street ทางคณะผู้จัดทำได้ทราบถึงปัญหาจากผู้ประกอบการ และจุดที่ต้องการให้เข้าไปพัฒนา เพื่อให้การบริหารจัดการสินค้าคงคลังนั้นมีประสิทธิภาพมากยิ่งขึ้น ดังนี้

1. พัฒนาระบบในการสนับสนุนการตัดสินใจการจัดการสินค้าคงคลัง จากรูปแบบของ Window Application
2. การจัดการข้อมูลต่างๆ ยังขาดการเชื่อมโยงข้อมูลถึงกันและไม่ได้ใช้ประโยชน์สูงสุดจากฐานข้อมูล
3. ไม่สามารถทราบถึงอัตราการหมุนเวียนของสินค้าแต่ละประเภทก่อให้เกิด Over Stock เพราะการสั่งไม่มีตัวสนับสนุนการตัดสินใจในการสั่งผลิต

### 3.1.4 การวิเคราะห์จุดบกพร่องการจัดการภายในองค์กร

จากข้อมูลองค์กรพบว่าการบริหารภายในนั้น จะใช้การจัดการข้อมูลด้วยระบบ Manual เกือบทั้งหมด ยกเว้นการจัดการสต็อกที่มีโปรแกรมเข้ามามีส่วนร่วม แต่โปรแกรมนั้นก็ไม่มีความสามารถในการสนับสนุนหรือช่วยในการตัดสินใจในการบริหารจัดการต่างๆ เพราะ ไม่สามารถวิเคราะห์ รูปแบบของกระแสความต้องการของลูกค้าได้ ทำให้คำสั่งการผลิตอาจเกิดความผิดพลาดได้ และขาดการเชื่อมโยงข้อมูลเพื่อนำมาใช้ให้ก่อประโยชน์สูงสุด

## 3.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับวิทยานิพนธ์ รูปแบบและเครื่องมือที่ใช้ในการแก้ไข้ปัญหา

### 3.2.1 ทฤษฎีที่เกี่ยวข้องกับวิทยานิพนธ์

#### 3.2.1.1 จุดสั่งซื้อกรณีความต้องการผันแปร (Reorder Point with Variable Demand)

ในการคำนวณจุดสั่งซื้อกับสต็อกเพื่อความปลอดภัยที่จะต้องอยู่ในระดับบริการ จะต้องกำหนดให้ความต้องการในแต่ละวันในช่วงเวลานำของการสั่งซื้อไม่แน่นอน เป็นอิสระต่อกัน และสามารถอธิบายได้ด้วย การกระจายแบบปกติ ค่าเฉลี่ยของความต้องการในช่วงเวลานำคือ ผลรวมของความต้องการเฉลี่ยในแต่ละวัน ในช่วงเวลานำ หรือก็คือผลคูณของค่าเฉลี่ยของความต้องการในแต่ละวันกับช่วงเวลานำ ( $\bar{d}L$ ) ส่วนความแปรปรวนของการกระจายก็คือ ผลรวมของความแปรปรวนในแต่ละวันสำหรับในช่วงเวลานำ ดังนั้นเราสามารถหาค่าของจุดสั่งซื้อในกรณีความต้องการผันแปรเพื่อให้อยู่ในระดับบริการที่ต้องการได้

### 3.2.1.2 ปริมาณสั่งซื้อกรณีต้องการผันแปร (Order Quantity with Variable Demand)

ถ้าอัตราความต้องการและช่วงเวลานำของการสั่งซื้อคงที่ ตัวแบบของระบบการสั่งซื้อตามช่วงเวลาจะกำหนดให้ปริมาณการสั่งซื้อคงที่ด้วย ซึ่งก็จะมีค่าเท่ากับปริมาณการสั่งอย่างประหยัด แต่ถ้าหากความต้องการผันแปร ปริมาณที่จะสั่งก็จะแตกต่างกันออกไป

ปริมาณการสั่งสำหรับตัวแบบช่วงเวลาการสั่งซื้อคงที่ที่มีความต้องการสินค้าแต่ละวันที่ผันแปรไปในลักษณะการกระจายแบบปกติ

### 3.2.1.3 อัตราการหมุนเวียนของสินค้าคงคลัง (Inventory Turnover)

อัตราการหมุนเวียนของสินค้า (Inventory Turnover) สามารถใช้เป็นตัวชี้การผลิตขององค์กร สามารถตอบสนองความต้องการของตลาดได้หรือไม่ และการบริหารสินค้าคงคลังได้ดีแค่ไหนเมื่อเทียบกับการขาย ในทางธุรกิจการเปลี่ยนวัตถุดิบและสินค้าเป็นเงินให้เร็วที่สุด มีส่วนสำคัญในทางธุรกิจทำให้สามารถบริหารงบการเงินขององค์กรเป็นไปอย่างมีประสิทธิภาพสูงสุด

## 3.2.2 รูปแบบของการพัฒนาระบบการสนับสนุนการตัดสินใจ

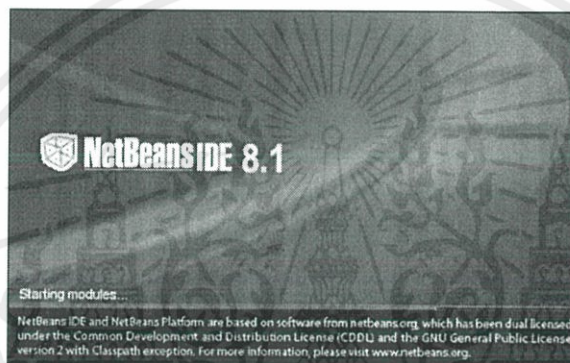
จากลักษณะรูปแบบของ Business Flow ข้อมูลการบริหารจัดการของบริษัท 7th Street และความต้องการของผู้ประกอบการ ทางคณะผู้จัดทำได้สังเกตเห็นถึงปัญหา และจุดที่สามารถจะเข้าไปพัฒนา เพื่อให้การบริหารจัดการสินค้าคงคลังนั้นมีประสิทธิภาพมากยิ่งขึ้น คือการสร้างระบบในการสนับสนุนการตัดสินใจการจัดการสินค้าคงคลัง ในรูปแบบของเว็บแอปพลิเคชัน เพราะ การจัดการข้อมูลต่างๆ ยังขาดการเชื่อมโยงข้อมูลถึงกันเพื่อใช้ประโยชน์สูงสุดจากฐานข้อมูล และเนื่องจากปัจจุบันอินเทอร์เน็ตมีความสะดวกสบายต่อการใช้งาน ง่ายต่อการเข้าถึง คณะผู้จัดทำจึงเลือกสร้างระบบสนับสนุนการตัดสินใจการจัดการสินค้าคงคลังผ่านระบบอินเทอร์เน็ต

### 3.2.2.1 เว็บแอปพลิเคชัน

เว็บแอปพลิเคชัน คือ การพัฒนาระบบงานบนเว็บ ข้อมูลต่างๆในระบบมีการไหลเวียนในแบบ Online ทั้งแบบ Local (ภายในวง LAN) และ Global (ออกไปยังเครือข่ายอินเทอร์เน็ต) ทำให้เหมาะสำหรับงานที่ต้องการข้อมูลแบบ Real Time ระบบมีประสิทธิภาพ และใช้งานง่าย ระบบสามารถเชื่อมโยงกับผู้ใช้บริการแบบ Real Time และรองรับข้อมูลที่จะถูกป้อนได้อย่าง Real Time จึงสามารถจะสร้างตัวช่วยสนับสนุนการตัดสินใจได้อย่างถูกต้องตามข้อมูลจริงในขณะนั้นๆได้อย่างมีประสิทธิภาพ

### 3.2.2.2 เครื่องมือที่ใช้ในการสร้างเว็บแอปพลิเคชัน

NetBeans IDE 8.1 คือ เครื่องมือสำหรับโปรแกรมเมอร์ที่จะใช้พัฒนาแอปพลิเคชันด้วยภาษา Java โดยที่โปรแกรม NetBeans นั้นเป็นโปรแกรมประเภท Open Source Software โดยผู้ใช้งานไม่จำเป็นต้องเสียเงิน เพื่อซื้อมาใช้งาน และยังเปิดเผย Source Code ให้ผู้สนใจและนักพัฒนานำไปดัดแปลง แก้ไข ตามกฎของ Open Source โดยมี Sun Micro System เป็นผู้สนับสนุนโครงการ ซึ่งตัวโปรแกรมแสดงดังรูปที่ 3.3



รูปที่ 3.3 ตัวโปรแกรม NetBeans

### 3.3 จัดการโครงสร้างฐานข้อมูลด้วยรูปแบบ ER Diagram และจัดเรียงข้อมูลผ่าน Microsoft Excel

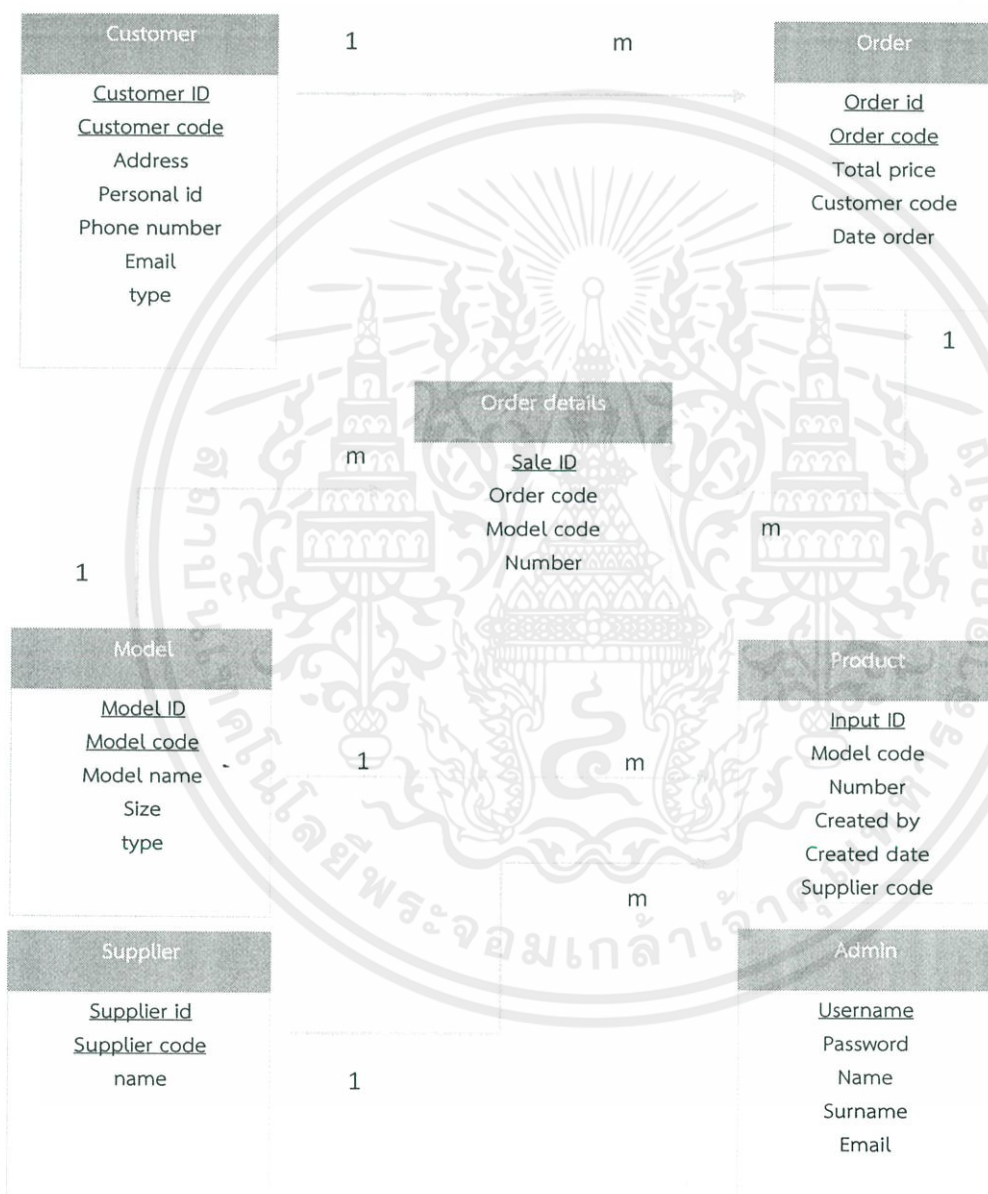
ER Diagram คือ แบบจำลองที่ใช้อธิบายโครงสร้างของฐานข้อมูลซึ่งเขียนออกมาในลักษณะของรูปภาพ การอธิบายโครงสร้างและความสัมพันธ์ของข้อมูล (Relationship) มีความสำคัญต่อการพัฒนาระบบงานฐานข้อมูล Application ต่างๆ ที่ต้องการการเก็บข้อมูลอย่างมีระบบ มีโครงสร้าง ดังนั้น ER Diagram จึงใช้เพื่อเป็นเอกสารในการสื่อสารระหว่าง นักออกแบบระบบ และนักพัฒนาระบบ เพื่อให้สื่อสารอย่างตรงกัน และเป็นสากลอีกด้วย

ประกอบด้วย

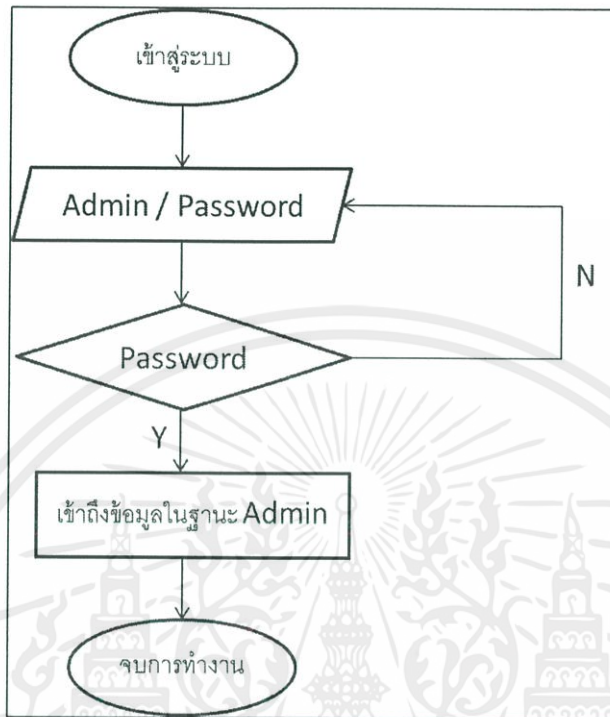
1. เอนทิตี (Entity) เป็นวัตถุ หรือสิ่งของที่เรานสนใจในระบบงานนั้น ๆ
2. แอททริบิว (Attribute) เป็นคุณสมบัติของวัตถุที่เรานสนใจ
3. ความสัมพันธ์ (Relationship) คือ ความสัมพันธ์ระหว่างเอนทิตี

### 3.3.1 สร้าง ER Diagram และFlow Chart

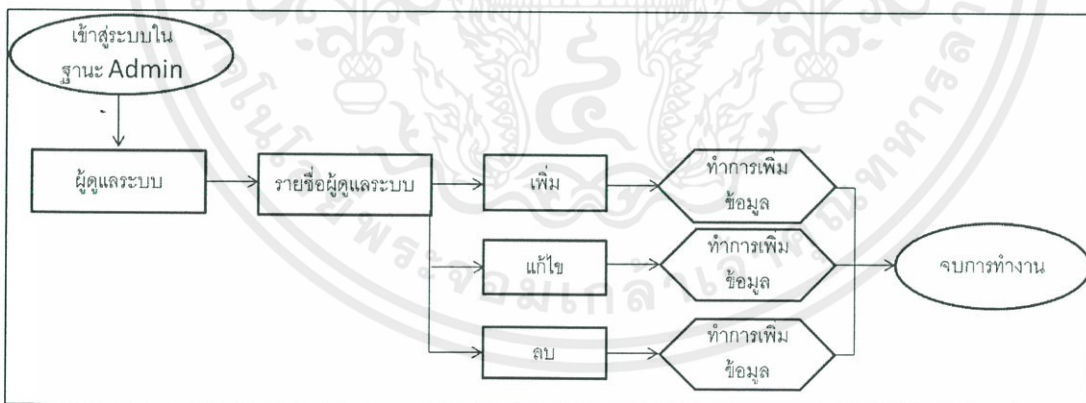
จากข้อมูลของสินค้าและองค์ประกอบต่างๆเพื่อเชื่อมโยงความสัมพันธ์ของข้อมูล และFlow Chart ดังรูปที่ 3.4-3.8



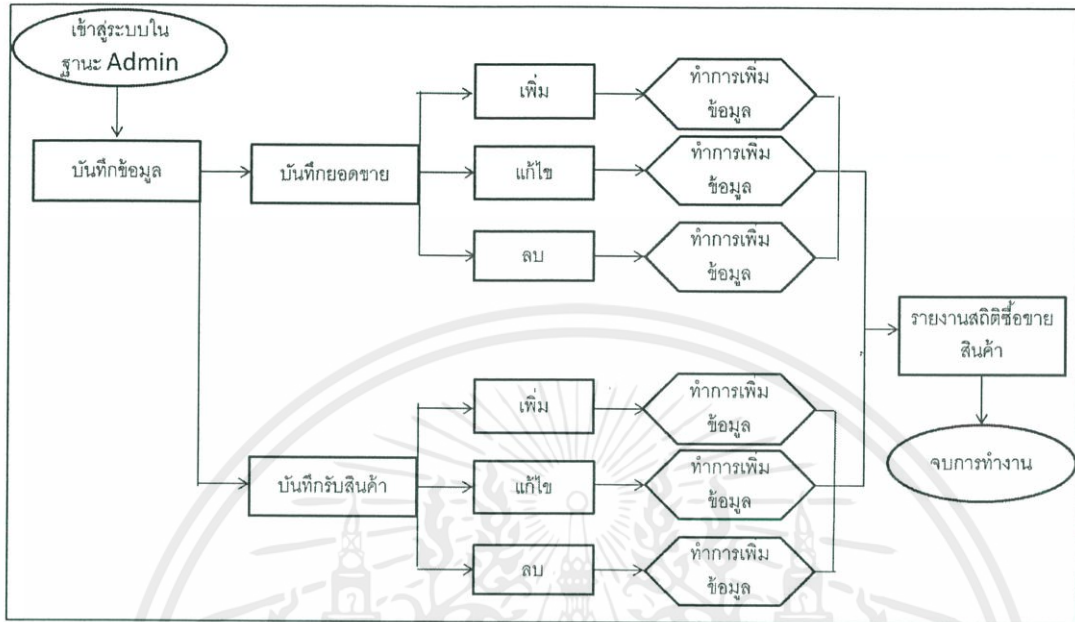
รูปที่ 3.4 ผังเชื่อมโยงความสัมพันธ์ของข้อมูล



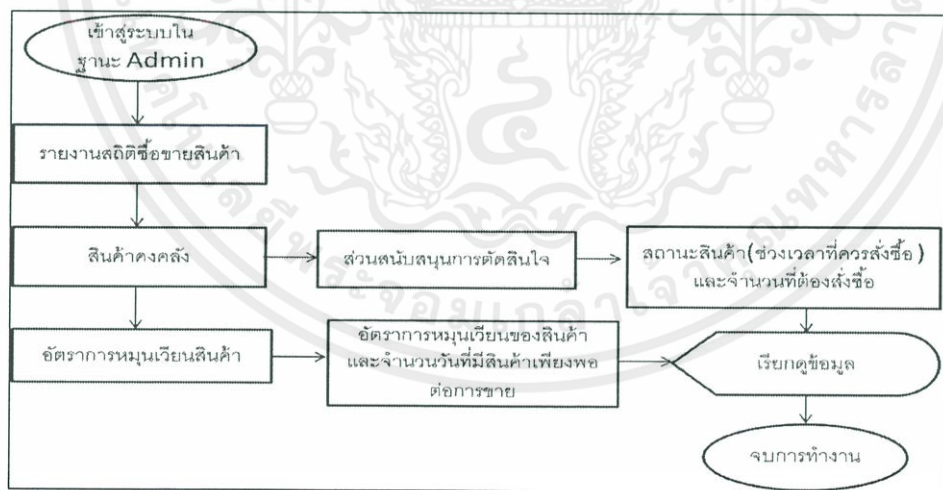
รูปที่ 3.5 แผนภาพการไหลของข้อมูล Flow Chart การเข้าใช้ระบบ



รูปที่ 3.6 แผนภาพการไหลของข้อมูล Flow Chart ผู้ดูแลระบบ



รูปที่ 3.7 แผนภาพการไหลของข้อมูล Flow Chart การใช้งานในส่วนของการบันทึกข้อมูลเป็นรายงานสถิติซื้อขายสินค้า



รูปที่ 3.8 แผนภาพการไหลของข้อมูล Flow Chart การใช้งานในส่วนของการสนับสนุนการตัดสินใจ และอัตราการหมุนเวียนสินค้า

พจนานุกรมข้อมูลแต่ละตาราง สร้างขึ้นเพื่อเป็นฐานข้อมูลโดยมีทั้งหมด 7 ตาราง (ดังตารางที่ 3.1 - ตารางที่ 3.7) โดยฟิลด์ที่แสดงคีย์ PK จะเป็นคีย์หลักของแต่ละตาราง ซึ่งเป็นตัวกำหนดให้ข้อมูลแต่ละเรคคอร์ดไม่ซ้ำกัน และรูปภาพที่ 3.9-3.14 แสดงตัวอย่างของข้อมูล

ตารางที่ 3.1 Admin

ลำดับ	ชื่อข้อมูล	ความหมาย	ประเภทข้อมูล	Key
1	admin_id	รหัสผู้ดูแลระบบ	Int	PK
2	username	ชื่อเข้าระบบผู้ดูแลระบบ	Varchar	
3	password	รหัสผ่านผู้ดูแลระบบ	Varchar	
4	name	ชื่อจริงผู้ดูแลระบบ	Varchar	
5	surname	นามสกุลผู้ดูแลระบบ	Varchar	
6	email	อีเมลผู้ดูแลระบบ	Varchar	

ตารางที่ 3.2 Model

ลำดับ	ชื่อข้อมูล	ความหมาย	ประเภทข้อมูล	Key
1	Model id	รหัสสินค้าแบบอัตโนมัติ	Int	
2	Model code	รหัสสินค้าแบบกำหนดเอง	Varchar	PK
3	Model name	ชื่อสินค้า	Varchar	
4	size	ไซส์สินค้า	Varchar	
5	type	ประเภทสินค้า	Varchar	

#	MODELID	MODELCODE	NAME	SIZE	TYPE
1		105 ST01012	เสื้อดำ	M	short
2		106 ST01013	เสื้อดำ	L	short
3		107 ST01014	เสื้อดำ	XL	short
4		205 ST01011	เสื้อดำ	S	short
5		206 ST01021	เสื้อขาว	S	short
6		207 ST01022	เสื้อขาว	M	short

รูปที่ 3.9 ตัวอย่างข้อมูลภายใน Model

ตารางที่ 3.3 Product

ลำดับ	ชื่อข้อมูล	ความหมาย	ประเภทข้อมูล	Key
1	Input id	รหัสอัตโนมัติ	Int	PK
2	Model code	รหัสสินค้า	Varchar	
3	number	จำนวนสินค้า	Int	
4	Created by	ชื่อผู้รับสินค้า	Varchar	
5	Created date	วันที่รับสินค้า	Varchar	
6	Supplier code	รหัสบริษัทที่จัดส่งสินค้า	Varchar	

#	INPUTID	CREATEDATE	CREATERBY	NUMBER	MODEL CODE	SUPPLIER CODE
1	6092017-05-31	state		200 ST01013		001
2	6102017-04-22	state		145 ST03013		002
3	6112017-07-31	state		400 ST03012		002
4	6122017-04-19	state		49 ST03012		001
5	6132017-05-31	state		250 ST01011		002
6	6142017-05-01	state		170 ST03014		001
7	6152017-03-31	state		155 ST03013		002
8	6162017-04-21	state		15 ST03013		001
9	6172017-04-26	state		51 ST02013		002
10	6182017-05-31	state		220 ST03013		002
11	6192017-04-21	state		9 ST02012		002
12	6202017-04-28	state		6 ST03012		001
13	6212017-05-31	state		100 ST02011		001
14	6222017-04-19	state		10 ST03011		002
15	6232017-04-17	state		7 ST02011		002
16	6242017-05-31	state		100 ST03014		002

รูปที่ 3.10 ข้อมูลภายใน Product

ตารางที่ 3.4 Supplier

ลำดับ	ชื่อข้อมูล	ความหมาย	ประเภทข้อมูล	Key
1	Supplier Id	รหัสอัตโนมัติ	Int	PK
2	Supplier code	รหัสบริษัทที่จัดส่งสินค้า	Varchar	
3	name	ชื่อบริษัทที่จัดส่งสินค้า	Varchar	

SUPPLIERID	NAME	SUPPCODE
1	1 สาขาใหม่	001
2	331.3PK	002

รูปที่ 3.11 ข้อมูลภายใน Supplier

ตารางที่ 3.5 Order

ลำดับ	ชื่อข้อมูล	ความหมาย	ประเภทข้อมูล	Key
1	Order id	รหัสอัตโนมัติ	Int	PK
2	Order code	รหัสคำสั่งซื้อ	Varchar	
3	Date order	วันที่จำหน่ายสินค้า	Varchar	
4	Custcode	ชื่อผู้ซื้อสินค้า	Varchar	

ORDERID	DATEORDER	ORDERCODE	CUSTCODE
1	324 2017-04-07	RC001	cust1
2	325 2017-04-08	RC002	cust1
3	326 2017-04-11	RC003	cust1
4	327 2017-04-13	RC004	cust1
5	328 2017-04-17	RC005	cust1
6	329 2017-04-18	RC006	cust1
7	330 2017-04-08	001	cust1
8	331 2017-04-09	002	cust1
9	332 2017-04-12	003	cust1
10	333 2017-04-08	004	cust2
11	334 2017-04-09	005	cust2
12	335 2017-04-10	006	cust2
13	336 2017-04-11	007	cust2
14	337 2017-04-12	008	cust2
15	338 2017-04-14	009	cust2
16	339 2017-04-15	010	cust2
17	340 2017-04-08	011	cust3
18	341 2017-04-09	012	cust3
19	342 2017-04-10	013	cust3
20	343 2017-04-11	014	cust3

รูปที่ 3.12 ข้อมูลภายใน Order

ตารางที่ 3.6 Order Detail

ลำดับ	ชื่อข้อมูล	ความหมาย	ประเภทข้อมูล	Key
1	Sale id	รหัสอัตโนมัติ	Int	
2	Sale code	ลำดับการสั่งซื้อ	Varchar	PK
3	number	จำนวนสินค้า	Int	
4	Model code	รหัสสินค้า	Varchar	
5	Order code	รหัสคำสั่งซื้อ	Varchar	

SELECT \* FROM DEMO.ORDDET X

#	SALEID	NUMBER	SALECODE	MODELCODE	ORDERCODE
1	716	10 1	ST01013	ST01013	2
2	717	10 1	ST01013	ST01013	3
3	718	10 1	ST01013	ST01013	4
4	719	10 1	ST01013	ST01013	5
5	740	10 1	ST01013	ST01013	6
6	741	10 1	ST01013	ST01013	7
7	742	10 1	ST01013	ST01013	8
8	743	10 1	ST01013	ST01013	9

รูปที่ 3.13 ข้อมูลภายใน Order Detail

ตารางที่ 3.7 Customer

ลำดับ	ชื่อข้อมูล	ความหมาย	ประเภทข้อมูล	Key
1	Id	รหัสอัตโนมัติ	Int	PK
2	Cust code	ชื่อผู้ซื้อสินค้า	Varchar	PK

SELECT \* FROM DEMO.CUSTOM X

#	ID	CUSTCODE
1	305	cust1
2	306	cust2
3	307	cust3
4	308	cust4
5	309	cust5
6	310	cust6
7	311	cust7
8	312	cust8

รูปที่ 3.14 ข้อมูลภายใน Customer

### 3.3.2 จัดเรียงข้อมูลผ่าน Microsoft Excel

นำข้อมูลที่ได้จากผู้ประกอบการ (ยอดการรับและจำหน่ายเดือนสิงหาคม 2559) มาจัดเรียงหมวดหมู่ เพื่อสะดวกต่อการนำไปใช้ในอนาคต ดังตารางที่ 3.8

ตารางที่ 3.8 ตัวอย่างข้อมูลการขายสินค้าเดือนสิงหาคม

Sale ID	Invoice ID	วันที่ขาย	สาขา	จุดค้า	หมวดหมู่	รหัสสินค้า	สินค้า	ราคาลด	จำนวน
POS 5650		01/09/2016 16:41	2		Tees (เสื้อยืด_แขนสั้น)	'0000050	7th street_ดำ-แดง_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000058	7th street_ดำ-เหลือง_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000078	7th street_ทอมบอท_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000098	Redsox_กรมเข้ม_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000122	Motor-X_ดำ_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000123	Motor-X_ดำ_L	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000126	Motor-X_กรม_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000127	Motor-X_กรม_L	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000130	Motor-X_กรมเข้ม_M	0	8
					Tees (เสื้อยืด_แขนสั้น)	'0000131	Motor-X_กรมเข้ม_L	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000180	ลูกศร_Runway_หม้อดำ_XL	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000202	DIOACTIVE_ดำ_M	0	20
					Tees (เสื้อยืด_แขนสั้น)	'0000213	NYC-เทฟ(เหลือง)_ขาว_5	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000223	เลข 7_ขาว_L	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000230	เลข 7_หม้อดำ_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000248	คาบดู_ขาว_XL	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000252	คาบดู_ดำ_XL	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000329	7ขลายทหาร_ดำ_S	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000330	7ขลายทหาร_ดำ_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000333	7ขลายทหาร_กรมเข้ม_S	0	7
					Tees (เสื้อยืด_แขนสั้น)	'0000334	7ขลายทหาร_กรมเข้ม_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000374	Raider_ดำ_M	0	10
					Tees (เสื้อยืด_แขนสั้น)	'0000376	Raider_กรมเข้ม_M	0	10

### 3.4 วางแผนการพัฒนาระบบผ่านรูปแบบของเว็บแอปพลิเคชันโดยโปรแกรม NetBeans IDE

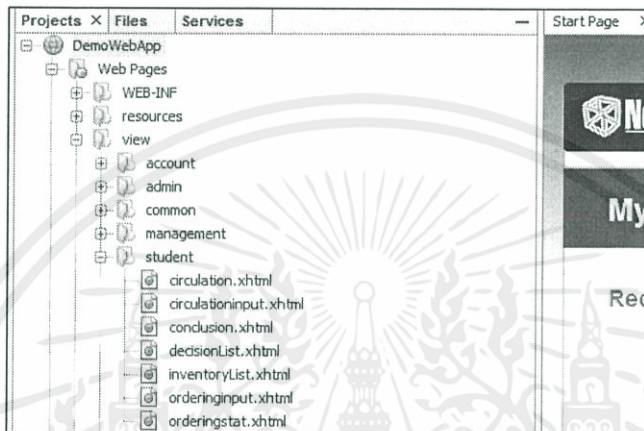
#### 3.4.1 เก็บความต้องการรูปแบบของหน้าจอแสดงผลจากผู้ประกอบการ

ตารางที่ 3.9 รูปแบบการแสดงผลที่ผู้ประกอบการต้องการเพื่อความง่ายต่อความเข้าใจ

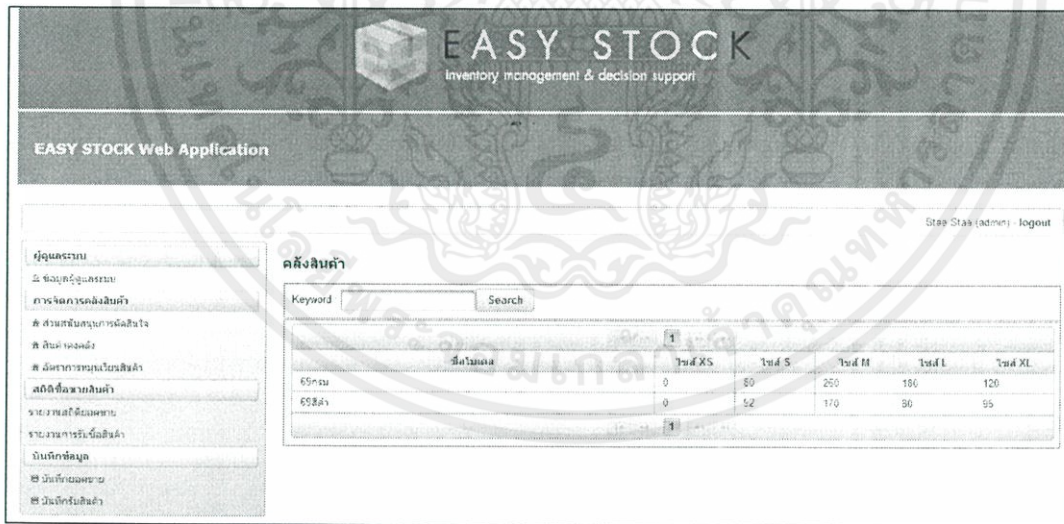
วัน	รหัสคำสั่งซื้อ	Modelcode	รหัสสินค้า	circulation				Summary
				e	m	l	ml	
1								0
2								0
3								0
4								0
5								0
6								0
7								0
8								0
9								0
10								0
11								0
12								0
13								0
14								0
15								0
16								0
17								0
18								0
19								0
20								0
21								0
22								0
23								0
24								0
25								0
26								0
27								0
28								0
29								0
30								0
31								0
32								0

### 3.4.2 ออกแบบหน้าจอ Mock Up ของหน้าตาเว็บแอปพลิเคชันในส่วนประกอบต่างๆ

หน้าจอ Mock Up ที่เป็นส่วนประกอบของหน้า Web ที่เราต้องการ ดังรูปที่ 3.15 ถึง 3.24 โดยการสร้างองค์ประกอบในส่วนของ Webpage/view/student/สร้างหน้าเพจตามที่ต้องการ (.xhtml)



รูปที่ 3.15 ส่วนประกอบของโปรเจกต์ในส่วนต่างๆ



รูปที่ 3.16 ส่วนของสถานะคลังสินค้า Inventory List

**EASY STOCK**  
inventory management & decision support

EASY STOCK Web Application

Site Site (admin) - logout

**รายงานการรับสินค้า**

Keyword:  Search

วันที่	บิลเลข	รหัสสินค้า	XS	S	M	L	XL
2017-04-01	001	เสื้อกรม	0	150	0	300	150
2017-04-01	001	เสื้อสีน้ำเงิน	0	0	450	0	150
2017-04-01	002	เสื้อกรม	0	0	300	0	0
2017-04-01	002	เสื้อสีน้ำเงิน	0	180	0	250	0

รูปที่ 3.17 ส่วนของรายละเอียดยอดสั่งซื้อของสินค้า

**EASY STOCK**  
inventory management & decision support

EASY STOCK Web Application

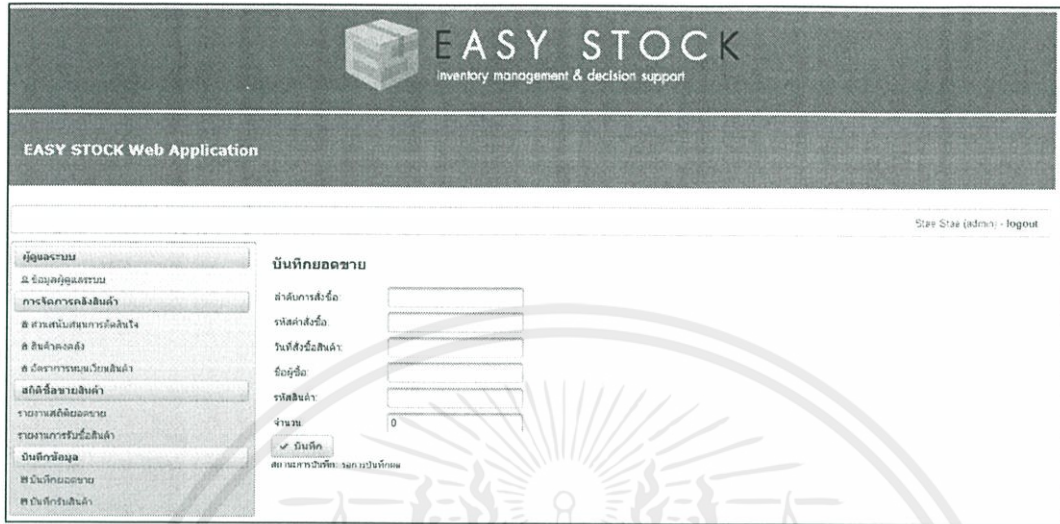
Site Site (admin) - logout

**สถิติยอดขายสินค้า**

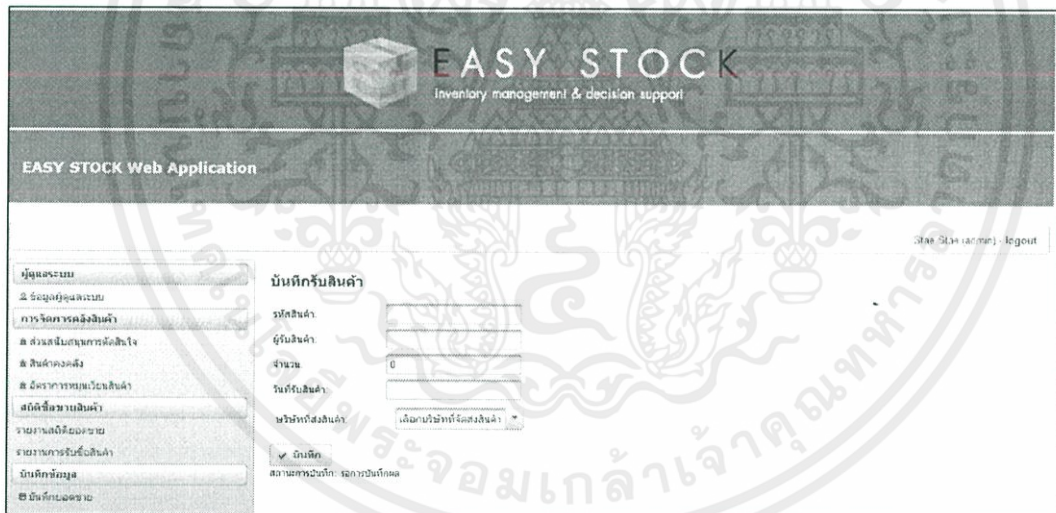
Keyword:  Search

วันที่	รหัสแจ้งซื้อ	ชื่อลูกค้า	รหัสสินค้า	XS	S	M	L	XL
2017-04-07	015	ศรีสวัสดิ์	เสื้อสีน้ำเงิน	0	0	20	0	0
2017-04-07	023	สมชาย	เสื้อสีน้ำเงิน	0	0	0	0	15
2017-04-07	ROS01	จเร:1	เสื้อกรม	0	20	20	10	0
2017-04-08	001	จเร:1	เสื้อกรม	0	0	0	0	10
2017-04-08	002	จเร:2	เสื้อสีน้ำเงิน	0	0	0	20	0
2017-04-08	011	จเร:2	เสื้อสีน้ำเงิน	0	0	70	0	0
2017-04-08	018	จเร:4	เสื้อสีน้ำเงิน	0	10	0	0	0

รูปที่ 3.18 ส่วนของรายละเอียดยอดขายสินค้า



รูปที่ 3.19 ส่วนของการบันทึกยอดขายสินค้าผ่านทางหน้าเว็บแอปพลิเคชัน



รูปที่ 3.20 ส่วนของการบันทึกยอดรับสินค้าผ่านทางหน้าเว็บแอปพลิเคชัน

**EASY STOCK**  
inventory management & decision support

EASY STOCK Web Application

Stee Staa (Admin) - logout

**ส่วนสนับสนุนการตัดสินใจ**

Keyword  Search

ชื่อสินค้า (model)	รหัสสินค้า	size	สถานะสินค้า	จำนวนคงเหลือ	จำนวนที่ต้องสั่งซื้อ
เสื้อดำ	ST01011	S	เสื้อแขนยาว # แขนงอศร มือ	92	123
เสื้อดำ	ST01012	M	เสื้อแขนยาว # แขนงอศร มือ	170	464
เสื้อดำ	ST01013	L	เสื้อแขนยาว # แขนงอศร มือ	83	315
เสื้อดำ	ST01014	XL	สินค้าต่อผล	95	0

รูปที่ 3.21 ส่วนของ Report ส่วนของการสนับสนุนการตัดสินใจ การจัดการคลังสินค้าของผู้ประกอบการ

**EASY STOCK**  
inventory management & decision support

EASY STOCK Web Application

Stee Staa (Admin) - logout

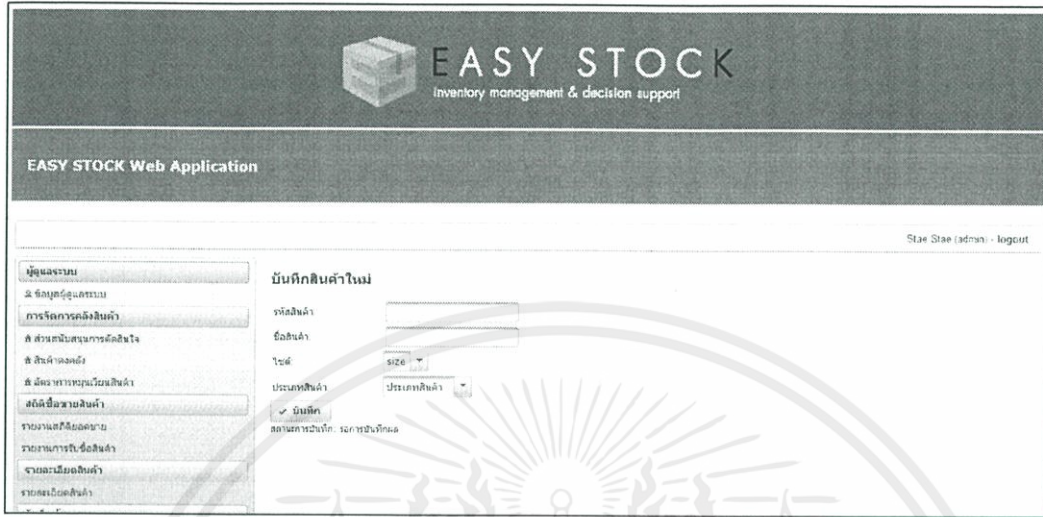
**อัตราการหมุนเวียนสินค้า**

Keyword  Search

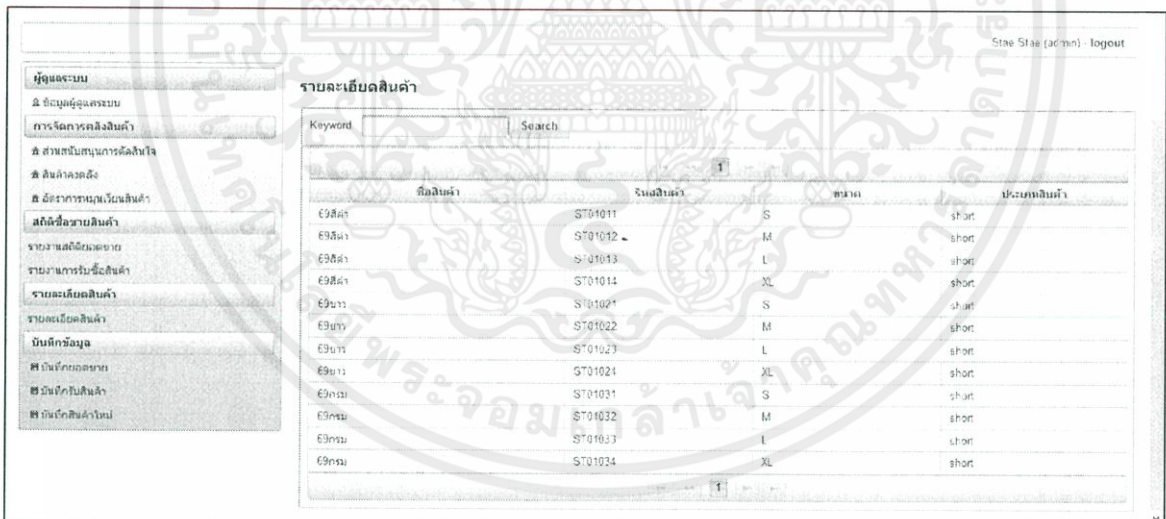
ชื่อสินค้า	รหัสสินค้า	size	Inventory turnover rate	Day of Supply
ST01011	เสื้อดำ	S	0.6479582205752113	10.818181818181818
ST01012	เสื้อดำ	M	0.907225364516123	7.75
ST01013	เสื้อดำ	L	1.0703970861930303	6.794117647058823
ST01014	เสื้อดำ	XL	0.4489757318367747	17.699500909090909
ST01031	เสื้อดำ	S	0.6089951521739131	11.5
ST01072	เสื้อดำ	M	0.4212924214242424	16.6

รูปที่ 3.22 ส่วนอัตราการหมุนเวียนสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 ส่วนของการบันทึกข้อมูลสินค้าเข้าสู่ฐานข้อมูล



รูปที่ 3.24 รายละเอียดสินค้าที่มีอยู่ในฐานข้อมูลในปัจจุบัน

### 3.4.3 การกำหนดรหัสของการจัดการสินค้าภายในคลัง

#### 3.4.3.1 กำหนดรหัสสินค้าเพื่อแยกประเภทของสินค้าแต่ละโมเดล

เพื่อระบุรหัสประจำสินค้านั้นๆ ให้สะดวกต่อการจัดการฐานข้อมูลที่ชัดเจน

XX - XX - XX - XX

XX : รหัสของของโมเดลสินค้า

XX : รหัสบกรุ่นสินค้า

XX : รหัสสี

XX : รหัสไซส์

ตารางที่ 3.11 รายละเอียดรหัสสินค้า

MODEL CODE	ITEMCODE	COLOR CODE	SIZE CODE
ST : แขนสั้น	01 : ลาย 69	01 : ดำ	S : 1
LG : แขนยาว	02 : แลบลายพราง	02 : ขาว	M : 2
	03 : 7 <sup>th</sup> Street	03 : กรม	L : 3
	04 : ลาย NEW YORK STAR	04 : เทา	XL : 4
	05 : ลายแลบลายพราง	05 : แดง	XXL : 5
	06 : ลายเลข 7	06 : กรมเข้ม	
	07 : 7 <sup>TH</sup> ลายทหาร	07 : เทาดิน	
	08 : ลาย SENSE	08 : เหลือง	
	09 : ลายท้อปซิดี้		
	10 : 7 <sup>th</sup> ARMY		

ตัวอย่าง : เสื้อแขนสั้นลาย69สีดำไซส์ M คือ ST69012n แยกข้อมูลประวัติการขายเดือนสิงหาคม ตามรูปแบบสินค้า จากข้อมูลการขายรวมของทางบริษัท (วันที่ 1-31 สิงหาคม 2559 )

### 3.4.3.2 กำหนดรหัสของออเดอร์สินค้า

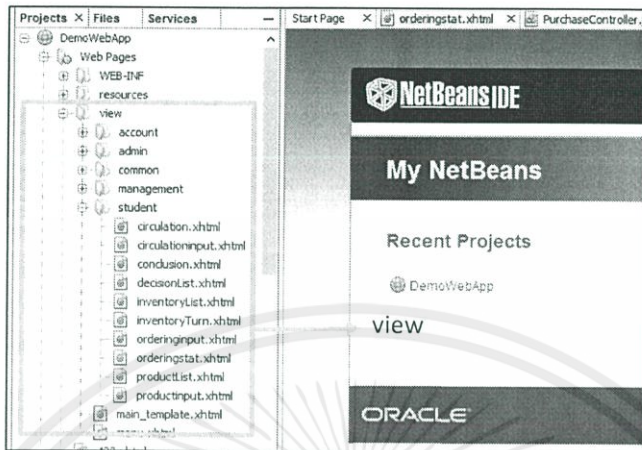
เพื่อระบุข้อมูล ให้ง่ายต่อการติดตามข้อมูลย้อนหลังว่าสินค้านั้นมาจากวันใด ใครเป็นผู้ซื้อ และมีรายการใดบ้างที่ถูกซื้อจากรหัสการซื้อนี้ เริ่มรหัสที่ 1 เป็นต้นไป

### 3.4.4 เขียนโปรแกรมทำให้ฟังก์ชันการทำงานภายใน

สามารถทำงานได้จริงและพัฒนา Report ให้ตรงกับความต้องการของผู้ประกอบการและเชื่อมโยงความสัมพันธ์ของข้อมูลเพื่อนำมาแสดงผลให้ตรงตามหมวดหมู่ของข้อมูล โดยเริ่มจากโครงสร้างการทำงานแบบ MVC จากที่กล่าวไปแล้วในบทที่ 2 ข้างต้น โดยเริ่มการสร้างส่วนต่างๆดังนี้

ส่วนที่ 1 ส่วน View เราจะสร้างส่วนประกอบดังนี้ โดยใช้เป็นประเภท .XHTML

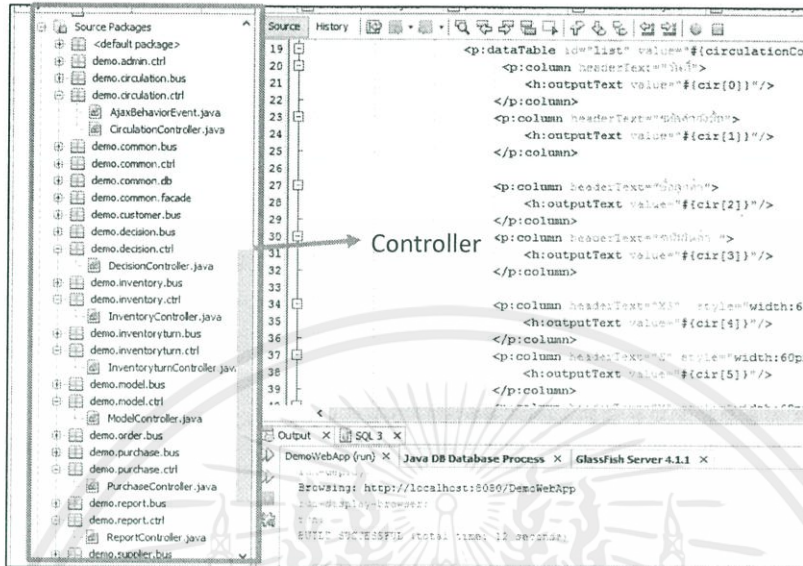
1. Circulation.xhtml
2. Circulationinput.xhtml
3. DecisionList.xhtml
4. InventoryList.xhtml
5. Inventoryturn.xhtml
6. Orderinginput.xhtml
7. Orderingstat.xhtml
8. ProductList.xhtml
9. Productinput.xhtml
10. Main\_template
11. Menu



รูปที่ 3.25 ส่วนประกอบของ View ที่จะแสดงผลต่อผู้ใช้งาน

ส่วนที่ 2 ส่วน Controller เราจะสร้างส่วนประกอบดังนี้ โดยใช้รูปแบบของ .Java

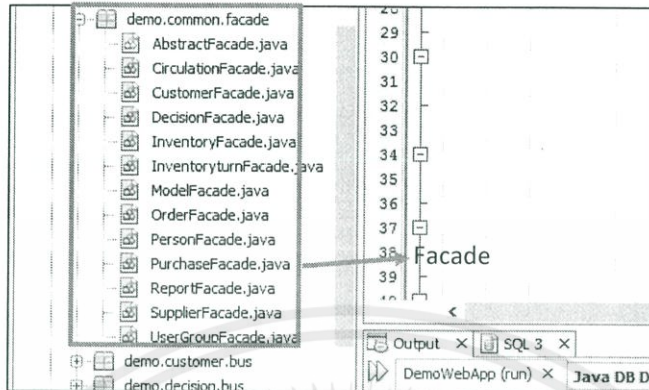
1. CirculationController.ctrl
2. DecisionController.ctrl
3. InventoryController.ctrl
4. InventoryturnController.ctrl
5. ModelController.ctrl
6. PurchaseController.ctrl



รูปที่ 3.26 ส่วนประกอบของ Controller

ส่วนที่ 3 ส่วน Model ใช้รูปแบบของ Java แบ่งเป็นสองส่วนคือ ส่วนของ Facade และ Service โดยFacade ประกอบไปด้วย

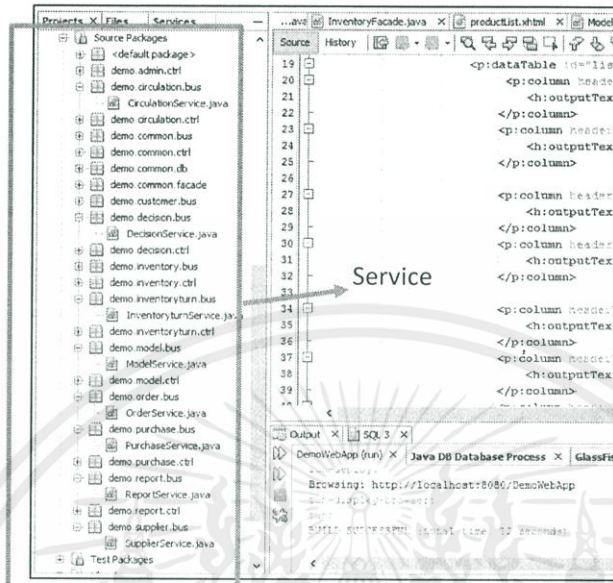
1. CirculationFacade.java
2. CustomerFacade.java
3. DecisionFacade.java
4. InventoryFacade.java
5. InventoryturnFacade.java
6. ModelFacade.java
7. OrderFacade.java
8. PurchaseFacade.java
9. SupplierFacade.java



รูปที่ 3.27 ส่วนประกอบภายใน Façade

ส่วนของ Service ประกอบไปด้วย

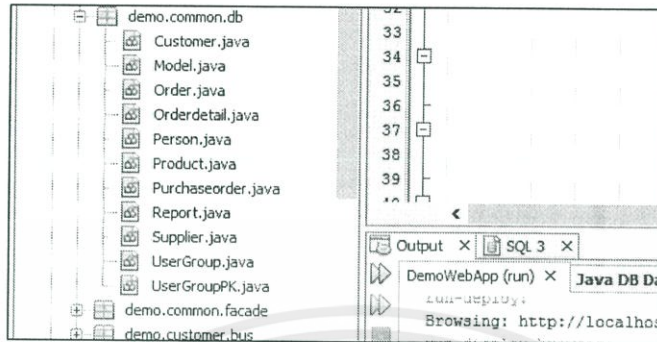
1. CirculationService.java
2. CustomerFacade.java
3. DecisionService.java
4. InventoryService.java
5. InventoryturnService.java
6. ModelService.java
7. OrderService.java
8. PurchaseService.java
9. SupplierService.java



รูปที่ 3.28 ส่วนประกอบภายใน Service

ส่วนที่ 4 ส่วนของ Database (demo.common.db) ประกอบไปด้วย

1. Customer.java
2. Model.java
3. Orderdetail.java
4. Order.java
5. Product.java
6. Purchaseorder.java
7. Supplier.java



รูปที่ 3.29 ส่วนประกอบภายใน Database

### 3.5 ทดสอบระบบการทำงานของเว็บแอปพลิเคชัน และสรุปผลการทำงานของระบบ

#### 3.5.1 ทดสอบระบบการทำงานของเว็บแอปพลิเคชัน

การทดลองการใช้งานในส่วนการทำงานต่างๆว่าสามารถใช้งานได้หรือไม่ โดยการทดสอบในส่วนต่างๆ ดังนี้ แล้วทำการบันทึกผล

##### 1. ด้านของคลังสินค้า

- ระบบสามารถแสดงปริมาณคลังสินค้าภายในคลังได้อย่างถูกต้อง
- สินค้าภายในคลังมีการเปลี่ยนแปลงเมื่อรับหรือจำหน่ายสินค้า

##### 2. การแสดงรายงานการรับ และจำหน่ายสินค้า ย้อนหลังได้ถูกต้องครบถ้วน

##### 3. ระบบการบันทึกข้อมูลการรับและจำหน่ายสินค้า

##### 4. อัตราการหมุนเวียนสินค้า

- สามารถแสดงอัตราการหมุนเวียนสินค้าได้
- สามารถแสดงค่าเพื่อการเปรียบเทียบสินค้าแต่ละประเภทได้
- แสดงค่าจำนวนวันที่มีเพียงพอต่อการขายได้

## 5. ส่วนสนับสนุนการตัดสินใจ

- สามารถแสดงสถานะความต้องการของสินค้า
- สามารถแสดงปริมาณที่ต้องทำการสั่งซื้อได้
- แสดงรายละเอียดต่างๆได้ถูกต้องครบถ้วน

### 3.5.2 ทดสอบประสิทธิภาพการสนับสนุนการตัดสินใจจากการทำงาน

ภายใต้มูลเดือนสิงหาคม 2559 ในรุ่น เสื้อยืดลาย 69 สีดำ และ ลาย 7th street สีดำ ลายละ 4 ไซส์ จำนวน 8 โมเดล โดยทำการป้อนข้อมูลแล้วดูความเปลี่ยนแปลงในส่วนต่างๆ และสังเกตการณ์เปลี่ยนแปลง ทำการบันทึกค่าการสนับสนุนการตัดสินใจให้ตรงตามารสนับสนุนการตัดสินใจจากเว็บแอปพลิเคชันแล้วทำการบันทึกผลเพื่อเปรียบเทียบ จำนวนสินค้าภายในคลังที่เกิดขึ้นระหว่างการสั่งซื้อเดิมของผู้ประกอบการ และการปริมาณคำสั่งซื้อจากการใช้ระบบสนับสนุนการตัดสินใจ เพื่อสังเกตความเปลี่ยนแปลงของปริมาณสินค้าภายในสินค้าคงคลัง

## บทที่ 4

### ผลการดำเนินงาน

#### 4.1 ศึกษาทฤษฎีที่เกี่ยวข้องกับปริญญานิพนธ์

ผู้วิจัยได้ทำการศึกษาข้อมูลและทฤษฎี เพื่อใช้ในการทำการออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่นในการนำเสนอผ่านรูปแบบของเว็บแอปพลิเคชัน โดยศึกษาจากผู้เชี่ยวชาญ หนังสือ และอินเทอร์เน็ต เพื่อนำข้อมูลต่างๆมาเชื่อมโยงความสัมพันธ์ และนำมาประยุกต์ใช้ในส่วนต่างๆของปริญญานิพนธ์ตามความเหมาะสม ให้เกิดความถูกต้องและสมบูรณ์มากที่สุด โดยได้รวบรวมไว้ในบทที่ 2

#### 4.2 สร้างระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น

ผลการศึกษาทดลองการออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น ประกอบไปด้วย 3 ส่วนด้วยกัน

1. ผลการออกแบบและพัฒนาเว็บแอปพลิเคชัน
2. การสร้างระบบการจัดการสินค้าคงคลัง
3. การสร้างส่วนสนับสนุนการตัดสินใจ (ปริมาณการสั่งซื้อ, สถานะความต้องการสินค้า)
4. การวิเคราะห์ข้อมูลจากการขายสินค้า
5. ทดสอบผลการใช้งานระบบสนับสนุนการตัดสินใจ

##### 4.2.1 ผลการออกแบบและพัฒนาเว็บแอปพลิเคชัน

จากการเก็บข้อมูลความต้องการจากผู้ประกอบการในด้านการใช้งานในส่วนประกอบต่างๆของเว็บแอปพลิเคชัน สิ่งที่พึงมีเพื่อเป็นประโยชน์แก่ผู้ประกอบการ และความสะดวกในการใช้งาน จึงออกมาเป็นในรูปแบบดังนี้

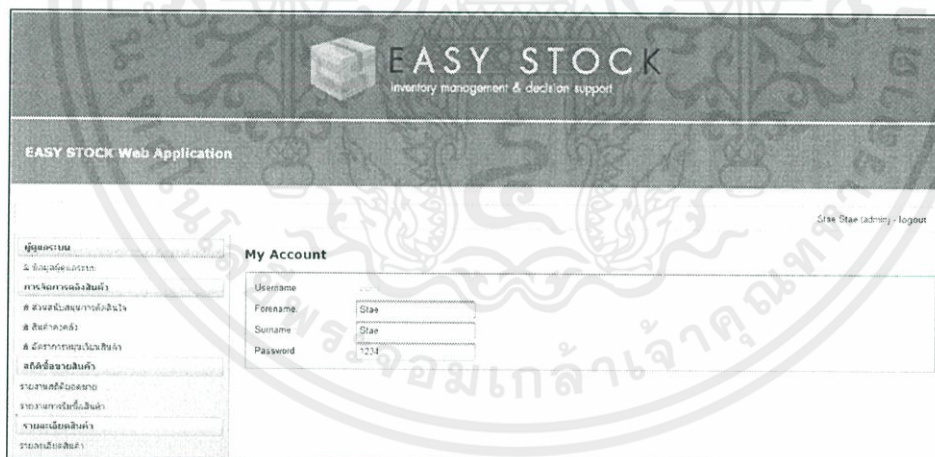
ส่วนที่ 1 หน้าหลักเพื่อทำการเข้าสู่ระบบการจัดการคลังสินค้าโดยผู้ดูแลระบบ ซึ่งเป็นผู้ที่สามารถเข้าไปดูและจัดการระบบภายในได้



รูปที่ 4.1 หน้าจอแสดงผลในส่วนของการระบบเพื่อเริ่มต้นการใช้งาน

ส่วนที่ 2 รูปแบบหน้าจอแสดงส่วนประกอบต่างๆของเว็บแอปพลิเคชัน

จะแสดงเมนูต่างๆที่พืงมีในระบบสนับสนุนการตัดสินใจของผู้ประกอบการ ในด้านซ้ายมือ เพื่อให้ผู้ใช้ได้เข้าใจการใช้งานได้ง่าย โดยแสดงส่วนต่างๆอย่างชัดเจนแยกเป็นประเภทของหัวข้อ และใช้ภาษาไทยเพื่ออำนวยความสะดวก เข้าใจ ประกอบไปด้วยหัวข้อหลักดังนี้: ผู้ดูแลระบบ, การจัดการคลังสินค้า, สถิติยอดขาย, การบันทึกข้อมูล



รูปที่ 4.2 หน้าจอโดยรวมของเว็บแอปพลิเคชัน

### ส่วนที่ 3 ส่วนของผู้ดูแลระบบ

จะเป็นส่วนที่แสดงถึงข้อมูลที่ทำให้การเข้าสู่ระบบว่าเป็นใครที่ทำให้การเข้าสู่ระบบนั้นมา ซึ่งจะขึ้นกับรหัสผ่านที่ทำให้การเข้าสู่ระบบมา เพื่อเป็นการยืนยันตัวตนของผู้ที่สามารถเข้ามาใช้งานระบบได้

My Account

Username	Stae
Forename	Stae
Surname	Stae
Password	1234

รูปที่ 4.3 ข้อมูลของผู้ดูแลระบบที่ทำให้การเข้าสู่ระบบมา

### ส่วนที่ 4 ส่วนของการค้นหาเพื่อตอบสนองการเรียกข้อมูลของผู้ใช้งาน รูปที่ 4.4

ค้นหาข้อมูล (ผลิตภัณฑ์)

Keyword	Search							
วันที่	รหัสคำสั่งซื้อ	ชื่อลูกค้า	รหัสสินค้า	XS	S	M	L	XL
No records found.								

รูปที่ 4.4 ส่วนของการค้นหาข้อมูลที่ต้องการ

#### 4.2.2 การสร้างระบบการจัดการสินค้าคงคลัง

ระบบสินค้าคงคลังนั้นมีทั้งการรับเข้าของสินค้าและการจำหน่ายออกของสินค้า ดังนั้นจึงทำสร้างระบบให้มีทั้งส่วนของการรับสินค้าเข้ามา และจำหน่ายสินค้าออกไป แสดงในรูปที่ 4.5 ส่วนของการบันทึกข้อมูล ประกอบไปด้วยสองส่วนคือ

ส่วนที่ 1 การบันทึกรับสินค้าเข้ามาในคลัง แสดงในรูปที่ 4.5 โดยจะต้องกรอกรายละเอียด คือ รหัสสินค้า ผู้รับสินค้า จำนวนสินค้า วันที่รับสินค้า และบริษัทที่ทำการจัดส่งสินค้า แล้วทำการบันทึก เมื่อทำการบันทึกข้อมูลแล้ว ข้อมูลจะไปแสดงในส่วนของสถิติการรับสินค้า เพื่อแสดงเป็นประวัติการรับสินค้า ทำให้สามารถตรวจสอบย้อนหลังได้ดังรูปที่ 4.6

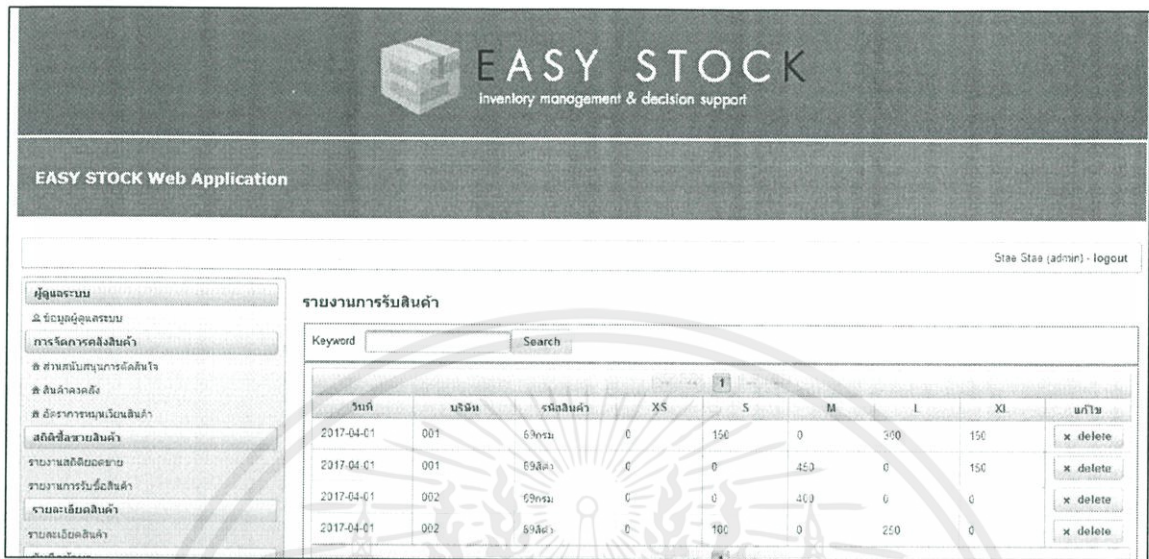
ผู้ดูแลระบบ

บันทึกรับสินค้า

ส่วนกรอกรายละเอียดการรับสินค้า

เมนูการบันทึกข้อมูล

รูปที่ 4.5 การบันทึกการรับสินค้าเข้าสู่คลัง



รูปที่ 4.6 รายงานการรับสินค้าหลังจากการบันทึกข้อมูลในส่วนของการบันทึกรับสินค้า จะมาแสดงเป็นข้อมูลย้อนหลังในส่วนสถิติซื้อขายสินค้า

ส่วนที่ 2 การบันทึกข้อมูลการจัดจำหน่ายสินค้า รูปที่ 4.7 โดยจะต้องกรอกข้อมูลต่างๆให้ครบถ้วน เพื่อเป็นข้อมูลการจัดจำหน่ายสินค้า เมื่อทำการบันทึกข้อมูลจะถูกนำไปแสดงในส่วนของ สถิติซื้อขายสินค้า หมวด รายงานสถิติยอดขาย ในรูปที่ 4.8 เพื่อแสดงข้อมูลย้อนหลังในการขายแต่ละครั้งได้

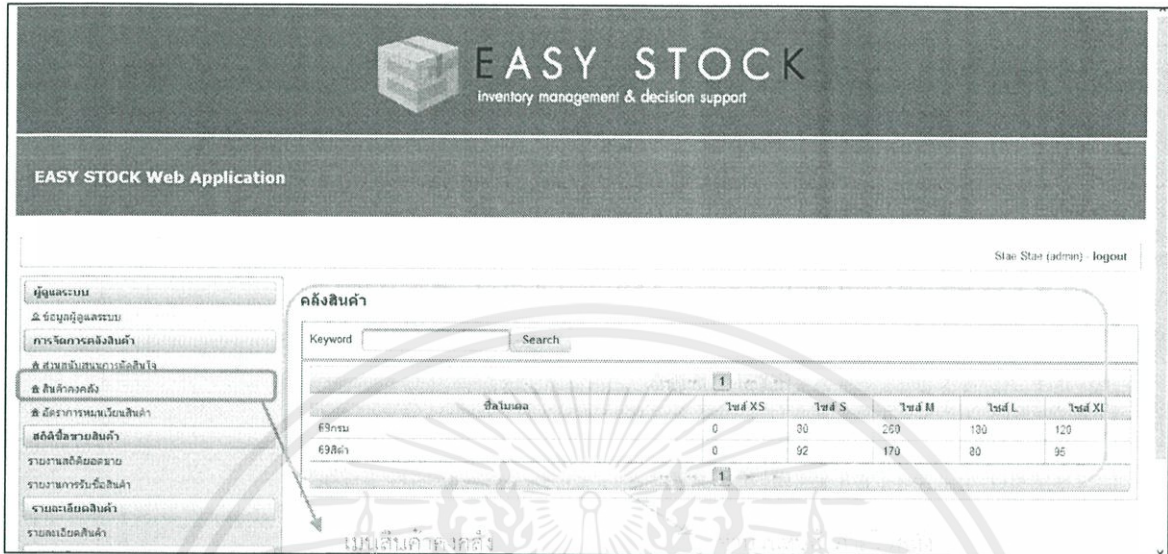
หลังจากสามารถบันทึกผลทั้งการรับสินค้าเข้า และจำหน่ายออกจากคลัง แล้วแสดงในส่วนของ รายงานสถิติซื้อขายสินค้านั้น ยังมีในส่วนของ สินค้าคงคลังที่ รูปที่ 4.9 จะเป็นส่วนที่แสดงยอดสุทธิภายใน คลังสินค้า ว่ามีสินค้านั้นใด ไซตใด จำนวนเท่าไรในเวลาปัจจุบัน เพราะเมื่อมีสินค้าเข้าออก ส่วนนี้จะทำการ อัปเดตผลทันที ให้เป็นสินค้าคงคลังจริงนะเวลานั้นๆ ทำให้ผู้ประกอบการสะดวกในการเช็คคลังสินค้าได้ ตลอดเวลา โดยที่ไม่ต้องตรวจนับภายในคลังสินค้า



รูปที่ 4.7 ส่วนการบันทึกข้อมูลการจัดจำหน่ายสินค้า และรายละเอียดที่ต้องกรอกเมื่อจำหน่ายสินค้า

วันที่	รหัสคำสั่งซื้อ	ชื่อลูกค้า	รหัสสินค้า	XS	S	M	L	XL	แก้ไข
2017-04-07	016	นางใจ	07สินค้า	0	6	20	0	0	x delete
2017-04-07	023	สมชาย	07สินค้า	0	0	0	0	15	x delete
2017-04-07	R0501	คุณตี	07กรม	0	20	20	10	0	x delete
2017-04-09	001	คุณตี	07กรม	0	0	0	0	10	x delete
2017-04-08	004	นางใจ	07สินค้า	0	0	0	20	0	x delete
2017-04-08	011	นางใจ	07สินค้า	0	0	20	0	0	x delete
2017-04-09	018	คุณตี	07สินค้า	0	10	0	0	0	x delete
2017-04-09	024	สมชาย	07สินค้า	0	0	0	0	10	x delete
2017-04-08	R0502	คุณตี	07กรม	0	20	50	60	0	x delete
2017-04-09	002	คุณตี	07กรม	0	0	0	0	10	x delete

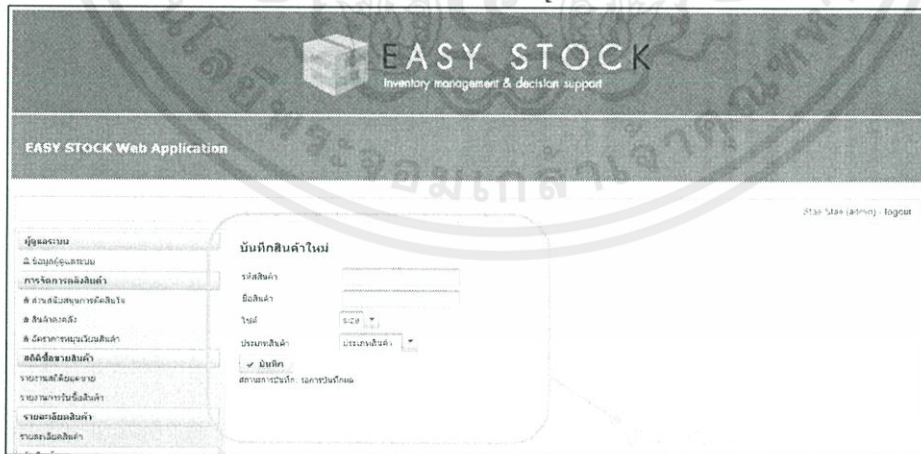
รูปที่ 4.8 รายงานสถิติยอดขายสินค้า



รูปที่ 4.9 จำนวนสินค้าภายในคลังสินค้า

#### 4.2.3 ส่วนการบันทึกและแสดงผลรายละเอียดสินค้า

เป็นส่วนที่เราทำการเพิ่มสินค้าเพื่อระบุข้อมูลของสินค้าเพื่อนำไปใช้ รูปที่ 4.10 และส่วนของการแสดงผลรายละเอียดต่างๆของสินค้า รูปที่ 4.11



รูปที่ 4.10 ส่วนของการบันทึกข้อมูลสินค้าเข้าสู่ฐานข้อมูล

Stae Stae (admin) - logout

**เมนูระบบ**

- ๕ ข้อมูลผู้ดูแลระบบ
- การจัดการคลังสินค้า
- ๓ ส่วนสนับสนุนการตัดสินใจ
- ๓ สินค้าคงคลัง
- ๓ อีเมลสารบบเว็บไซต์
- ๓ สถิติการขายสินค้า
- รายงานสถิติยอดขาย
- รายงานการรับสินค้า
- รายละเอียดสินค้า**
- รายละเอียดสินค้า
- บันทึกข้อมูล
- ๓ บันทึกยอดขาย
- ๓ บันทึกรับสินค้า
- ๓ บันทึกสินค้าใหม่

**รายละเอียดสินค้า**

Keyword:  Search

ชื่อสินค้า	รหัสสินค้า	ขนาด	ประเภทสินค้า
ยีสต์ดำ	ST01011	S	short
ยีสต์ดำ	ST01012	M	short
ยีสต์ดำ	ST01013	L	short
ยีสต์ดำ	ST01014	XL	short
ยีสต์ขาว	ST01021	S	short
ยีสต์ขาว	ST01022	M	short
ยีสต์ขาว	ST01023	L	short
ยีสต์ขาว	ST01024	XL	short
ยีสต์กรม	ST01031	S	short
ยีสต์กรม	ST01032	M	short
ยีสต์กรม	ST01033	L	short
ยีสต์กรม	ST01034	XL	short

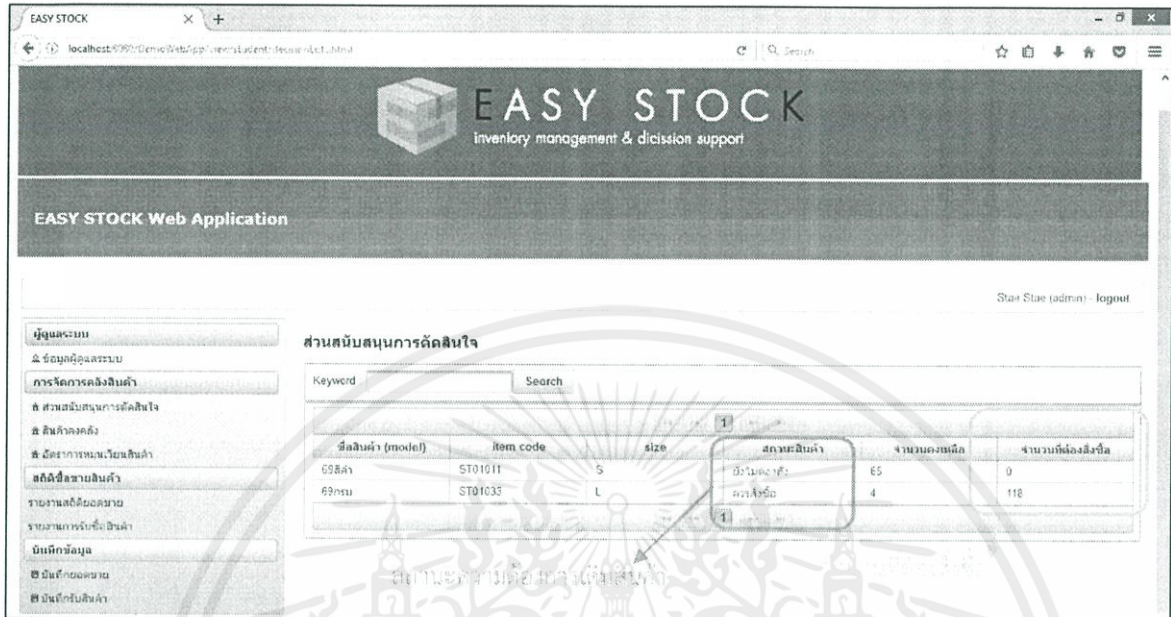
รูปที่ 4.11 รายละเอียดสินค้าที่มีอยู่ในฐานข้อมูลในปัจจุบัน

#### 4.2.4 การสร้างส่วนสนับสนุนการตัดสินใจ

ส่วนสนับสนุนการตัดสินใจ รูปที่ 4.12 นั้นจะประกอบไปด้วยข้อมูลต่างๆที่เป็นประโยชน์ต่อผู้ประกอบการซึ่งจะแสดงเป็นสัดส่วนที่มีความง่ายต่อการเข้าใจ โดยที่ผู้ประกอบการไม่ต้องคิดเอง โดยใช้การคำนวณจากสถิติยอดขายย้อนหลัง เพื่อช่วยในการตัดสินใจปริมาณการสั่งซื้อ ช่วงเวลาที่ควรสั่งซื้อ ซึ่งแสดงในสถานะความต้องการเติมสินค้า โดยจะแบ่งเป็นสามระยะ คือ

- 1.ระยะปลอดภัยคือระยะที่ยังไม่ต้องทำการสั่งซื้อสินค้าเข้าคลัง
- 2.ระยะเฝ้าระวังคือระยะที่สินค้าใกล้ที่จะต้องตรวจสอบเพื่อสั่งซื้อ
- 3.คือระยะที่ต้องสั่งซื้อสินค้าทันที และนอกจากนี้ข้อดีของส่วนนี้คือสามารถอัพเดทข้อมูลได้

ตลอดเวลาเมื่อมีการรับหรือจำหน่ายสินค้าออกไปทำให้รู้ถึงความเคลื่อนไหวของสถานะสินค้า ณ เวลาปัจจุบันได้



รูปที่ 4.12 ส่วนข้อมูลที่เป็นประโยชน์ในการตัดสินใจซื้อสินค้าของผู้ประกอบการ ประกอบไปด้วย สถานะความต้องการเพิ่มสินค้า จำนวนคงเหลือในปัจจุบัน และจำนวนที่ต้องสั่งซื้อ

#### 4.2.5 การวิเคราะห์ข้อมูลจากการขายสินค้า

สร้างส่วนวิเคราะห์สินค้าในแต่ละสินค้าจากการคำนวณจากยอดขายสินค้านั้น เพื่อใช้ประโยชน์ในการดูว่าสินค้านั้นมีการตอบรับเป็นอย่างไร สินค้าที่ได้ซื้อเข้ามาเก็บไว้นั้นมีการหมุนเวียนของสินค้ามากเพียงใด เพื่อนำไปเป็นประโยชน์ในอนาคตหากต้องการดูกระแสของสินค้า ว่าสินค้าใดควรจัดโปรโมชั่นเพื่อทำการกระตุ้นยอดขายสินค้าใดควรลดหรือเลิกการผลิต สินค้าใดควรสั่งเพิ่มในอนาคตเพราะผลตอบรับดี โดยได้ใส่ในส่วนของ อัตราการหมุนเวียนของสินค้า และจำนวนวันที่มีสินค้าเพียงพอต่อการขาย (Day of Supply) ดังแสดงในรูปที่ 4.13

EASY STOCK Web Application

Sta: Sta (admin) - logout

**ผู้ดูแลระบบ**

- ผู้รับผิดชอบผู้ดูแลระบบ
- การจัดการคลังสินค้า**
- ดู ส่วนสัมพันธ์ของการคลังสินค้า
- ดู สินค้าคงคลัง
- ดู อัตราการหมุนเวียนสินค้า
- สถิติการขายสินค้า
- รายงานคลังสินค้า
- รายงานการรับซื้อสินค้า
- รายละเอียดสินค้า
- รายละเอียดสินค้า
- บันทึกข้อมูล
- บันทึกยอดขาย
- บันทึกรับสินค้า
- บันทึกใบสั่งซื้อ

**อัตราการหมุนเวียนสินค้า**      จำนวนที่สินค้ามีเพียงพอต่อการขาย

Keyword:  Search

รหัสสินค้า	ชื่อสินค้า	size	Inventory turnover rate	Day of Supply
ST01011	เสื้อสีม่วง	S	61.71	34.0
ST01012	เสื้อสีฟ้า	M	90.32	24.36
ST01013	เสื้อสีฟ้า	L	103.03	21.35
ST01014	เสื้อสีฟ้า	XL	41.9	49.0
ST01031	เสื้อกรม	S	60.67	36.14
ST01032	เสื้อกรม	M	42.42	51.85
ST01033	เสื้อกรม	L	50.0	44.0
ST01034	เสื้อกรม	XL	22.22	99.0

รูปที่ 4.13 การวิเคราะห์สินค้าจากข้อมูลการขายสินค้า

#### 4.2.6 ทดสอบการใช้งานระบบสนับสนุนการตัดสินใจ

จากการทดสอบการใช้งานของโปรแกรม โดยใช้ขอบเขต เสื้อสองรุ่น รุ่นละสี่ขนาด ทำให้มีชุดข้อมูลทั้งหมด 8 โมเดลด้วยกันที่นำมาทดสอบการใช้งานของโปรแกรมว่าสามารถใช้งานได้จริงหรือไม่ และทำการสังเกตผลที่ได้ โดยมีตัวชี้วัดผลการทดสอบในหัวข้อต่างๆดังต่อไปนี้

ตัวชี้วัด

การทำงาน

1. ด้านของคลังสินค้า (รูปที่ 4.14)
  - ระบบสามารถแสดงปริมาณคลังสินค้าภายในคลังได้อย่างถูกต้อง : สามารถทำงานได้
  - สินค้าภายในคลังมีการเปลี่ยนแปลงเมื่อรับหรือจำหน่ายสินค้า : สามารถทำงานได้
2. การแสดงรายงานการรับ และจำหน่ายสินค้า ย้อนหลังได้ : สามารถทำงานได้
3. ระบบการบันทึกข้อมูลการรับและจำหน่ายสินค้า (รูปที่ 4.16) : สามารถทำงานได้
4. อัตราการหมุนเวียนสินค้า (รูปที่ 4.17)
  - สามารถแสดงอัตราการหมุนเวียนสินค้าได้ : สามารถทำงานได้
  - สามารถแสดงค่าเพื่อการเปรียบเทียบสินค้าแต่ละประเภทได้ : สามารถทำงานได้
  - แสดงค่าจำนวนวันที่มีเพียงพอต่อการขายได้ : สามารถทำงานได้

5. ส่วนสนับสนุนการตัดสินใจ (รูปที่ 4.18)

- สามารถแสดงสถานะความต้องการของสินค้าได้ : สามารถทำงานได้
- สามารถแสดงปริมาณที่ต้องทำการสั่งซื้อได้ : สามารถทำงานได้
- แสดงรายละเอียดต่างๆได้ถูกต้องครบถ้วน : สามารถทำงานได้

**คลังสินค้า**

Keyword  Search

ชื่อในเกณฑ์	ไซส์ XS	ไซส์ S	ไซส์ M	ไซส์ L	ไซส์ XL
50กรม	0	20	250	180	120
60ลำ	0	55	170	80	55

รูปที่ 4.14 สินค้าคงคลัง

**สต็อกยอดขายสินค้า**

Keyword  Search

วันที่	ยอดคงเหลือ	ชื่อลูกค้า	จริงใบแจ้ง	XS	S	M	L	XL
2017-04-07	C15	นายพร	60ลำ	0	0	20	0	0
2017-04-07	C23	สมชาย	60ลำ	0	0	0	0	15
2017-04-07	R02021	นาย1	50กรม	0	20	20	10	0
2017-04-08	C01	นาย1	50กรม	0	0	0	0	10
2017-04-08	C04	นาย2	60ลำ	0	0	0	20	0
2017-04-08	C11	นาย3	60ลำ	0	0	70	0	0
2017-04-08	C18	นาย4	60ลำ	0	10	0	0	0
2017-04-08	C24	สมชาย	60ลำ	0	0	0	0	10
2017-04-08	R0502	นาย1	50กรม	0	30	50	60	0
2017-04-09	C02	นาย1	50กรม	0	0	0	0	10
2017-04-09	C05	นาย2	50ลำ	0	0	0	50	0
2017-04-09	C12	นาย3	60ลำ	0	0	50	0	0
2017-04-09	C19	นาย4	60ลำ	0	20	0	0	0
2017-04-09	C25	สมชาย	60ลำ	0	0	0	0	10
2017-04-10	C06	นาย2	60ลำ	0	0	0	5	0
2017-04-10	C13	นาย3	60ลำ	0	0	10	0	0
2017-04-10	C20	นาย4	60ลำ	0	10	0	0	0
2017-04-10	C25	สมชาย	60ลำ	0	0	0	0	5
2017-04-11	C07	นาย2	50ลำ	0	0	0	25	0

รูปที่ 4.15 รายงานสต็อกยอดการจัดจำหน่ายย้อนหลัง

**รายงานการรับสินค้า**

Keyword:  Search

วันที่	บริษัท	ชนิดสินค้า	XS	S	M	L	XL
2017-04-01	001	เสื้อกรม	0	150	0	300	150
2017-04-01	001	เสื้อสีดำ	0	0	450	0	150
2017-04-01	002	เสื้อกรม	0	0	400	0	0
2017-04-01	002	เสื้อสีดำ	0	100	0	250	0

รูปที่ 4.16 รายงานสถิติการรับสินค้าย้อนหลัง

**อัตราการผลิตหมุนเวียนสินค้า**

Keyword:  Search

ชนิดสินค้า	สีสินค้า	size	Inventory turnover rate	Day of Supply
ST01011	เสื้อสีดำ	S	0.518161818181818182	11.323529411764707
ST01012	เสื้อสีดำ	M	0.3032250054516129	7.75
ETC1013	เสื้อสีดำ	L	1.0333030303030303	5.7941176470566323
ST01014	เสื้อสีดำ	XL	0.4439795918367347	15.559090909090909
ST01031	เสื้อกรม	S	0.503555521735431	11.5
ST01032	เสื้อกรม	M	0.42424242424242425	16.5
ST01033	เสื้อกรม	L	0.4439795918367347	15.559090909090909
ST01034	เสื้อกรม	XL	0.2222222222222222	31.5

รูปที่ 4.17 ตารางแสดงอัตราการหมุนเวียนสินค้าและ จำนวนวันที่มีสินค้าเพียงพอต่อการขายจากรางจะเห็นได้ว่าเมื่อแสดงจำนวนตัวเลขขึ้นมา ทำให้สามารถทราบสินค้าได้มีการหมุนเวียนมากกว่ากัน

ส่วนสนับสนุนการตัดสินใจ

Keyword  Search

ชื่อสินค้า (model)	รหัสสินค้า	size	สถานะสินค้า	จำนวนคงเหลือ	จำนวนที่จองสั่งซื้อ
เสื้อผ้า	STD1011	S	เสร็จทั้งหมด !! และเร่งการ ผลิต	85	112
เสื้อผ้า	STD1012	M	เสร็จทั้งหมด !! และเร่งการ ผลิต	170	484
เสื้อผ้า	STD1013	L	เสร็จทั้งหมด !! และเร่งการ ผลิต	80	316
เสื้อผ้า	STD1014	XL	สินค้าเพียงพอ	85	0
เสื้อขน	STD1031	S	เสร็จทั้งหมด !! และเร่งการ ผลิต	80	102
เสื้อขน	STD1032	M	เฝ้าระวัง	260	105
เสื้อขน	STD1033	L	เฝ้าระวัง	190	114
เสื้อขน	STD1034	XL	สินค้าเพียงพอ	120	0

รูปที่ 4.18 ตารางส่วนของการสนับสนุนการตัดสินใจ (สถานะความต้องการเติมสินค้า, ปริมาณที่ต้องสั่งซื้อ)

#### 4.2.7 การเปรียบเทียบประสิทธิภาพในการทำงาน

จากการทดสอบการใช้งานของโปรแกรม โดยใช้ขอบเขต เสื้อสองรุ่น รุ่นละสี่ขนาด ในระยะเวลา 1 เดือน ทำให้มีชุดข้อมูลทั้งหมด 8 โมเดลด้วยกันที่นำมาทดสอบการใช้งานของโปรแกรมว่ามีประสิทธิภาพการทำงานดีขึ้นหรือไม่ และทำการสังเกตผลที่ได้ ดังต่อไปนี้

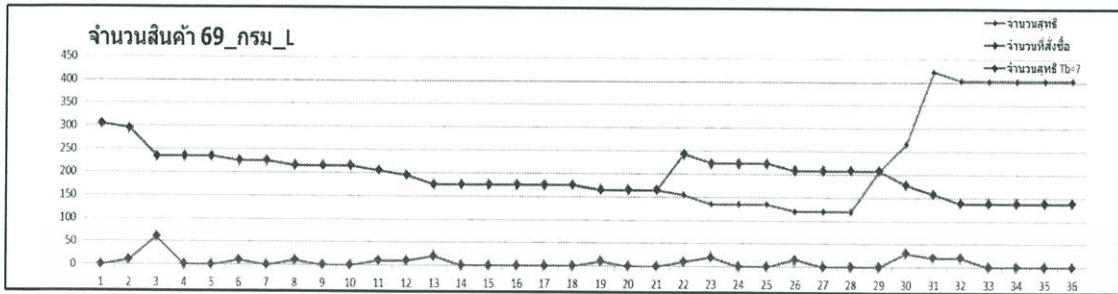
1. การปรับปรุงประสิทธิภาพสินค้ารุ่น 69 กรัม L จำนวนความต้องการสินค้าต่อวัน และจำนวนสินค้าคงคลังสุทธิจากผู้ประกอบการกับโปรแกรม

ตารางที่ 4.1 สินค้ารุ่น 69 กรัม L

เริ่มคงคลังที่ 300	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
69_กรัม_L	1	305	305	0
69_กรัม_L	2	295	295	10
69_กรัม_L	3	235	235	60
69_กรัม_L	4	235	235	0
69_กรัม_L	5	235	235	0
69_กรัม_L	6	225	225	10
69_กรัม_L	7	225	225	0
69_กรัม_L	8	215	215	10
69_กรัม_L	9	215	215	0
69_กรัม_L	10	215	215	0
69_กรัม_L	11	205	205	10
69_กรัม_L	12	195	195	10
69_กรัม_L	13	175	175	20
69_กรัม_L	14	175	175	0
69_กรัม_L	15	175	175	0

ตารางที่ 4.1 สิ้นค้ารุ่น 69 กรัม L (ต่อ)

เริ่มคลังที่ 300	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ ตั้งชื่อ
69_กรม_L	16	175	175	0
69_กรม_L	17	175	175	0
69_กรม_L	18	175	175	0
69_กรม_L	19	165	165	10
69_กรม_L	20	165	165	0
69_กรม_L	21	165	165	0
69_กรม_L	22	155	243	10
69_กรม_L	23	135	223	20
69_กรม_L	24	135	223	0
69_กรม_L	25	135	223	0
69_กรม_L	26	120	208	15
69_กรม_L	27	120	208	0
69_กรม_L	28	120	208	0
69_กรม_L	29	206	208	0
69_กรม_L	30	266	178	30
69_กรม_L	31	423	158	20
69_กรม_L	32	403	138	20
69_กรม_L	33	403	138	0
69_กรม_L	34	403	138	0
69_กรม_L	35	403	138	0



รูปที่ 4.19 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 69 กรัม L

จากตารางที่ 4.1 และรูปที่ 4.19 แสดงการเปรียบเทียบยอดของสินค้าคงคลังว่ามีจำนวนสินค้าคงคลังเหลือปริมาณเท่าใดภายใต้เงื่อนไข reorder point และ วิเคราะห์ระบบสินค้าคงคลังด้วยตัวแบบสินค้าคงคลังเชิงสโตคาสติก มีดังนี้

1.1 การเปรียบเทียบระหว่างวิธีการใช้ข้อมูลจริงกับข้อมูลจากโปรแกรมเพื่อดูผลของสินค้าที่อยู่ในคลังมากเกินไปของโมเดลสินค้า 69 กรัม L พบว่าหลังจากการใช้เว็บแอปพลิเคชัน การสนับสนุนการตัดสินใจแล้ว สินค้า 69 กรัม L ต้องมีการสั่งจำนวน 1 ครั้ง คือ วันที่ 15 สิงหาคม จำนวน 78 ตัว ซึ่งจะได้รับสินค้าในวันที่ 22 สิงหาคม โดยมีแนวโน้มที่แตกต่างจากการสั่งซื้อเดิมที่สั่งวันที่ 22 สิงหาคม จำนวน 86 ตัว, 23 สิงหาคม จำนวน 60 ตัว, และ 24 สิงหาคม จำนวน 157 ตัว ซึ่งจะได้รับสินค้าในวันที่ 29 สิงหาคม, 30 สิงหาคม, และ 31 สิงหาคม ซึ่งสามารถจัดการลดสินค้าคงคลังได้จำนวน 65.76 เปอร์เซ็นต์

1.2 การเปรียบเทียบค่าพื้นที่ใต้กราฟ จากพื้นที่ใต้กราฟของข้อมูลจริงมีค่าพื้นที่ใต้กราฟเท่ากับ 8,224 หน่วย\*วัน และข้อมูลจากโปรแกรมมีค่าพื้นที่ใต้กราฟเท่ากับ 7,031.5 หน่วย\*วัน ซึ่งผลแตกต่างเท่ากับ 1,192.5 หน่วย\*วัน

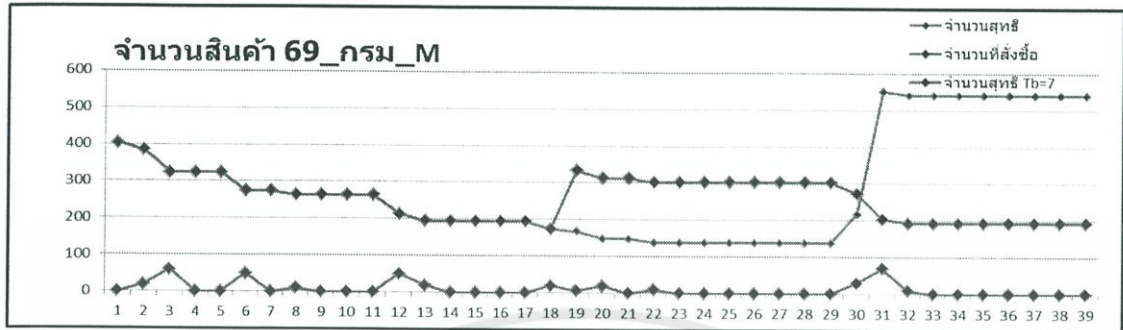
2. การปรับปรุงประสิทธิภาพสินค้ารุ่น 69 กรัม M จำนวนความต้องการสินค้าต่อวัน และจำนวนสินค้าคงคลังสุทธิจากผู้ประกอบการกับโปรแกรม

ตารางที่ 4.2 สินค้ารุ่น 69 กรัม M

เริ่มคลังที่ 400	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
69_กรัม_M	1	404	404	0
69_กรัม_M	2	384	384	20
69_กรัม_M	3	324	324	60
69_กรัม_M	4	324	324	0
69_กรัม_M	5	324	324	0
69_กรัม_M	6	274	274	50
69_กรัม_M	7	274	274	0
69_กรัม_M	8	264	264	10
69_กรัม_M	9	264	264	0
69_กรัม_M	10	264	264	0
69_กรัม_M	11	264	264	0
69_กรัม_M	12	214	214	50
69_กรัม_M	13	194	194	20
69_กรัม_M	14	194	194	0
69_กรัม_M	15	194	194	0
69_กรัม_M	16	194	194	0
69_กรัม_M	17	194	194	0
69_กรัม_M	18	174	174	20

ตารางที่ 4.2 สิ้นค้ารุ่น 69 กรัม M (ต่อ)

เริ่มครั้งที่ 400	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
69_กรัม_M	19	168	333	6
69_กรัม_M	20	148	313	20
69_กรัม_M	21	148	313	0
69_กรัม_M	22	138	303	10
69_กรัม_M	23	138	303	0
69_กรัม_M	24	138	303	0
69_กรัม_M	25	138	303	0
69_กรัม_M	26	138	303	0
69_กรัม_M	27	138	303	0
69_กรัม_M	28	138	303	0
69_กรัม_M	29	138	303	0
69_กรัม_M	30	218	273	30
69_กรัม_M	31	551	203	70
69_กรัม_M	32	541	193	10
69_กรัม_M	33	541	193	0
69_กรัม_M	34	541	193	0
69_กรัม_M	35	541	193	0
69_กรัม_M	36	541	193	0



รูปที่ 4.20 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 69 กรัม M

จากตารางที่ 4.2 และรูปที่ 4.20 แสดงการเปรียบเทียบยอดของสินค้าคงคลังว่ามีจำนวนสินค้าคงคลังเหลือปริมาณเท่าใดภายใต้เงื่อนไข reorder point และ วิเคราะห์ระบบสินค้าคงคลังด้วยตัวแบบสินค้าคงคลังเชิงสโตคาสติก มีดังนี้

2.1 การเปรียบเทียบระหว่างวิธีการใช้ข้อมูลจริงกับข้อมูลจากโปรแกรมเพื่อดูผลของสินค้าที่อยู่ในคลังมากเกินไปของโมเดลสินค้า 69 กรัม M พบว่าหลังจากการใช้เว็บแอปพลิเคชัน การสนับสนุนการตัดสินใจแล้ว สินค้า 69 กรัม M ต้องมีการสั่งจำนวน 1 ครั้ง คือ วันที่ 12 สิงหาคม จำนวน 159 ตัว ซึ่งจะได้รับสินค้าในวันที่ 19 สิงหาคม โดยมีแนวโน้มที่แตกต่างจากการสั่งซื้อเดิมที่สั่งวันที่ 23 สิงหาคม จำนวน 80 ตัว, และ 24 สิงหาคม จำนวน 333 ตัว ซึ่งจะได้รับสินค้าในวันที่ 30 สิงหาคม, และ 31 สิงหาคม ซึ่งสามารถจัดการลดสินค้าคงคลังได้จำนวน 64.33 เปอร์เซ็นต์

2.2 การเปรียบเทียบค่าพื้นที่ใต้กราฟ จากพื้นที่ใต้กราฟของข้อมูลจริงมีค่าพื้นที่ใต้กราฟเท่ากับ 10,914.5 หน่วย\*วัน และข้อมูลจากโปรแกรมมีค่าพื้นที่ใต้กราฟเท่ากับ 9,826.5 หน่วย\*วัน ซึ่งผลแตกต่างเท่ากับ 1,192.5 หน่วย\*วัน

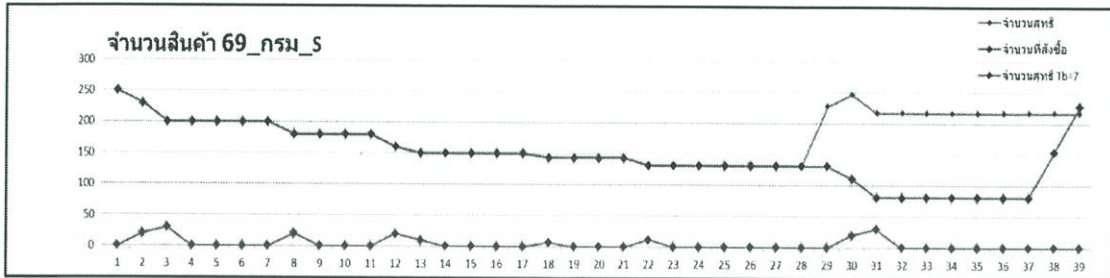
3. การปรับปรุงประสิทธิภาพสินค้ารุ่น 69 กรัม S จำนวนความต้องการสินค้าต่อวัน และจำนวนสินค้าคงคลังสุทธิจากผู้ประกอบการกับโปรแกรม

ตารางที่ 4.3 สินค้ารุ่น 69 กรัม S

เริ่มคงคลังที่	วันที่	จำนวนสุทธิ	จำนวนสุทธิ	จำนวนที่
250			Tb=7	สั่งซื้อ
69_กรัม_S	1	250	250	0
69_กรัม_S	2	230	230	20
69_กรัม_S	3	200	200	30
69_กรัม_S	4	200	200	0
69_กรัม_S	5	200	200	0
69_กรัม_S	6	200	200	0
69_กรัม_S	7	200	200	0
69_กรัม_S	8	181	180	20
69_กรัม_S	9	181	180	0
69_กรัม_S	10	181	180	0
69_กรัม_S	11	181	180	0
69_กรัม_S	12	161	160	20
69_กรัม_S	13	151	150	10
69_กรัม_S	14	151	150	0
69_กรัม_S	15	151	150	0
69_กรัม_S	16	151	150	0
69_กรัม_S	17	151	150	0
69_กรัม_S	18	144	143	7

ตารางที่ 4.3 สีนคาร์บอน 69 กรัม S (ต่อ)

เริ่มคคครั้งที่ 250	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
69_กรม_S	19	144	143	0
69_กรม_S	20	144	143	0
69_กรม_S	21	144	143	0
69_กรม_S	22	132	131	12
69_กรม_S	23	132	131	0
69_กรม_S	24	132	131	0
69_กรม_S	25	132	131	0
69_กรม_S	26	132	131	0
69_กรม_S	27	132	131	0
69_กรม_S	28	132	131	0
69_กรม_S	29	227	131	0
69_กรม_S	30	247	111	20
69_กรม_S	31	217	81	30
69_กรม_S	32	217	81	0
69_กรม_S	33	217	81	0
69_กรม_S	34	217	81	0
69_กรม_S	35	217	81	0
69_กรม_S	36	217	81	0
69_กรม_S	37	217	81	0
69_กรม_S	38	217	154	0



รูปที่ 4.21 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 69 กรม S

จากตารางที่ 4.3 และรูปที่ 4.21 แสดงการเปรียบเทียบยอดของสินค้าคงคลังว่ามีจำนวนสินค้าคงคลังเหลือปริมาณเท่าใดภายใต้เงื่อนไข reorder point และ วิเคราะห์ระบบสินค้าคงคลังด้วยตัวแบบสินค้าคงคลังเชิงสโตคาสติก มีดังนี้

3.1 การเปรียบเทียบระหว่างวิธีการใช้ข้อมูลจริงกับข้อมูลจากโปรแกรมเพื่อผลของสินค้าที่อยู่ในคลังมากเกินไปของโมเดลสินค้า 69 กรม S พบว่าหลังจากการใช้เว็บแอปพลิเคชัน การสนับสนุนการตัดสินใจแล้ว สินค้า 69 กรม S ต้องมีการสั่งจำนวน 1 ครั้ง คือ วันที่ 31 สิงหาคม จำนวน 73 ตัว ซึ่งจะได้รับสินค้าในวันที่ 7 กันยายน โดยมีแนวโน้มที่แตกต่างจากการสั่งซื้อเดิมที่สั่งวันที่ 22 สิงหาคม จำนวน 95 ตัว, และ 23 สิงหาคม จำนวน 20 ตัว ซึ่งจะได้รับสินค้าในวันที่ 29 สิงหาคม, และ 30 สิงหาคม ซึ่งสามารถจัดการลดสินค้าคงคลังได้จำนวน 29.03 เปอร์เซนต์

3.2 การเปรียบเทียบค่าพื้นที่ใต้กราฟ จากพื้นที่ใต้กราฟของข้อมูลจริงมีค่าพื้นที่ใต้กราฟเท่ากับ 6,813.5 หน่วย\*วัน และข้อมูลจากโปรแกรมมีค่าพื้นที่ใต้กราฟเท่ากับ 5,550.5 หน่วย\*วัน ซึ่งผลแตกต่างเท่ากับ 1,263 หน่วย\*วัน

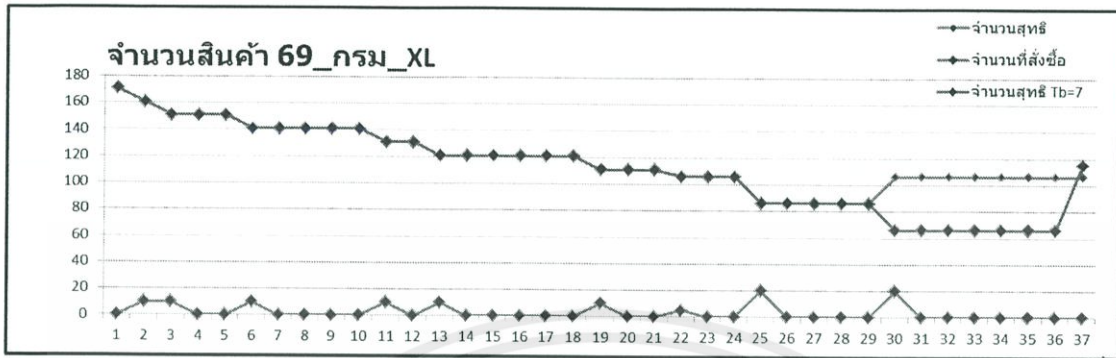
4. การปรับปรุงประสิทธิภาพสินค้ารุ่น 69 XL จำนวนความต้องการสินค้าต่อวัน และจำนวนสินค้าคงคลังสุทธิจากผู้ประกอบการกับโปรแกรม

ตารางที่ 4.4 สินค้ารุ่น 69 กรัม XL

เริ่มคงคลังที่ 170	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
69_กรัม_XL	1	171	171	0
69_กรัม_XL	2	161	161	10
69_กรัม_XL	3	151	151	10
69_กรัม_XL	4	151	151	0
69_กรัม_XL	5	151	151	0
69_กรัม_XL	6	141	141	10
69_กรัม_XL	7	141	141	0
69_กรัม_XL	8	141	141	0
69_กรัม_XL	9	141	141	0
69_กรัม_XL	10	141	141	0
69_กรัม_XL	11	131	131	10
69_กรัม_XL	12	131	131	0
69_กรัม_XL	13	121	121	10
69_กรัม_XL	14	121	121	0
69_กรัม_XL	15	121	121	0
69_กรัม_XL	16	121	121	0
69_กรัม_XL	17	121	121	0
69_กรัม_XL	18	121	121	0
69_กรัม_XL	19	111	111	10

ตารางที่ 4.4 สิ้นค้ารุ่น 69 กรัม XL (ต่อ)

เริ่มคงคลังที่ 170	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
69_กรัม_XL	20	111	111	0
69_กรัม_XL	21	111	111	0
69_กรัม_XL	22	106	106	5
69_กรัม_XL	23	106	106	0
69_กรัม_XL	24	106	106	0
69_กรัม_XL	25	86	86	20
69_กรัม_XL	26	86	86	0
69_กรัม_XL	27	86	86	0
69_กรัม_XL	28	86	86	0
69_กรัม_XL	29	86	86	0
69_กรัม_XL	30	106	66	20
69_กรัม_XL	31	106	66	0
69_กรัม_XL	32	106	66	0
69_กรัม_XL	33	106	66	0
69_กรัม_XL	34	106	66	0
69_กรัม_XL	35	106	66	0
69_กรัม_XL	36	106	66	0
69_กรัม_XL	37	106	115	0



รูปที่ 4.22 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 69 กรม XL

จากตารางที่ 4.4 และรูปที่ 4.22 แสดงการเปรียบเทียบยอดของสินค้าคงคลังว่ามีจำนวนสินค้าคงคลังเหลือปริมาณเท่าใดภายใต้เงื่อนไข reorder point และ วิเคราะห์ระบบสินค้าคงคลังด้วยตัวแบบสินค้าคงคลังเชิงสโตคาสติก มีดังนี้

4.1 การเปรียบเทียบระหว่างวิธีการใช้ข้อมูลจริงกับข้อมูลจากโปรแกรมเพื่อดูผลของสินค้าที่อยู่ในคลังมากเกินไปของโมเดลสินค้า 69 กรม XL พบว่าหลังจากการใช้เว็บแอปพลิเคชัน การสนับสนุนการตัดสินใจแล้ว สินค้า 69 กรม XL ต้องมีการสั่งจำนวน 2 ครั้ง คือ วันที่ 30 สิงหาคม จำนวน 49 ตัว, และวันที่ 31 สิงหาคม จำนวน ซึ่งจะได้รับสินค้าในวันที่ 49 กันยายน โดยมีแนวโน้มที่แตกต่างจากการสั่งซื้อเดิมที่สั่งวันที่ 23 สิงหาคม จำนวน 20 ตัว ซึ่งจะได้รับสินค้าในวันที่ 30 สิงหาคม ซึ่งสามารถจัดการลดสินค้าคงคลังได้จำนวน 6.91 เปอร์เซ็นต์

4.2 การเปรียบเทียบค่าพื้นที่ใต้กราฟ จากพื้นที่ใต้กราฟของข้อมูลจริงมีค่าพื้นที่ใต้กราฟเท่ากับ 4,374.5 หน่วย\*วัน และข้อมูลจากโปรแกรมมีค่าพื้นที่ใต้กราฟเท่ากับ 4,132.5 หน่วย\*วัน ซึ่งผลแตกต่างเท่ากับ 242 หน่วย\*วัน

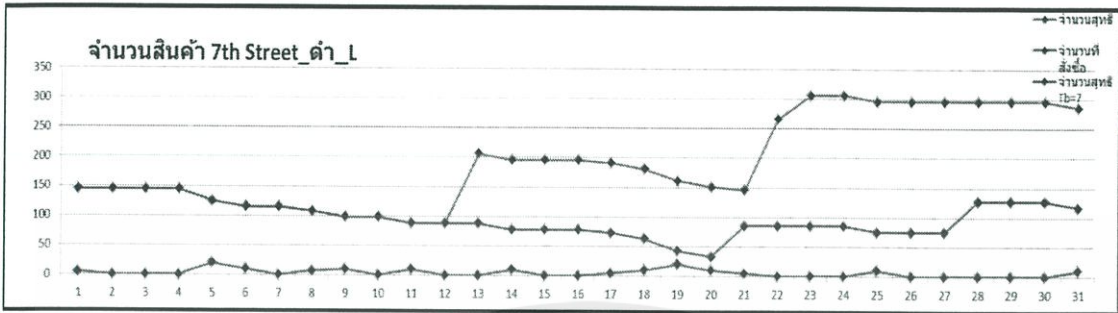
5. การปรับปรุงประสิทธิภาพสินค้ารุ่น 7th Street ดำ L จำนวนความต้องการสินค้าต่อวัน และจำนวนสินค้าคงคลังสุทธิจากผู้ประกอบการกับโปรแกรม

ตารางที่ 4.5 สินค้ารุ่น 7th Street ดำ L

เริ่มคลังที่ 150	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
7th Street ดำ_L	1	145	145	5
7th Street ดำ_L	2	145	145	0
7th Street ดำ_L	3	145	145	0
7th Street ดำ_L	4	145	145	0
7th Street ดำ_L	5	125	125	20
7th Street ดำ_L	6	115	115	10
7th Street ดำ_L	7	115	115	0
7th Street ดำ_L	8	108	108	7
7th Street ดำ_L	9	98	98	10
7th Street ดำ_L	10	98	98	0
7th Street ดำ_L	11	88	88	10
7th Street ดำ_L	12	88	88	0
7th Street ดำ_L	13	206	88	0
7th Street ดำ_L	14	196	78	10
7th Street ดำ_L	15	196	78	0
7th Street ดำ_L	16	196	78	0
7th Street ดำ_L	17	191	73	5
7th Street ดำ_L	18	181	63	10
7th Street ดำ_L	19	161	43	20

ตารางที่ 4.5 สินค้ารุ่น 7th Street ดำ L (ต่อ)

เริ่มคลังที่ 150	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
7th Street_ดำ_L	20	151	33	10
7th Street_ดำ_L	21	146	85	5
7th Street_ดำ_L	22	266	85	0
7th Street_ดำ_L	23	306	85	0
7th Street_ดำ_L	24	306	85	0
7th Street_ดำ_L	25	296	75	10
7th Street_ดำ_L	26	296	75	0
7th Street_ดำ_L	27	296	75	0
7th Street_ดำ_L	28	296	127	0
7th Street_ดำ_L	29	296	127	0
7th Street_ดำ_L	30	296	127	0
7th Street_ดำ_L	31	286	117	10



รูปที่ 4.23 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 7th Street ดำ L

จากตารางที่ 4.5 และรูปที่ 4.23 แสดงการเปรียบเทียบยอดของสินค้าคงคลังว่ามีจำนวนสินค้าคงคลังเหลือปริมาณเท่าใดภายใต้เงื่อนไข reorder point และ วิเคราะห์ระบบสินค้าคงคลังด้วยตัวแบบสินค้าคงคลังเชิงสโตคาสติก มีดังนี้

5.1 การเปรียบเทียบระหว่างวิธีการใช้ข้อมูลจริงกับข้อมูลจากโปรแกรมเพื่อดูผลของสินค้าที่อยู่ในคลังมากเกินไปของโมเดลสินค้า 7th Street ดำ L พบว่าหลังจากการใช้เว็บแอปพลิเคชัน การสนับสนุนการตัดสินใจแล้ว สินค้า 7th Street ดำ L ต้องมีการสั่งจำนวน 2 ครั้ง คือ วันที่ 14 สิงหาคม จำนวน 15 ตัว และ 21 สิงหาคม จำนวน 51 ตัว ซึ่งจะได้รับสินค้าในวันที่ 21 สิงหาคม และ 28 สิงหาคม โดยมีแนวโน้มที่แตกต่างจากการสั่งซื้อเดิมที่สั่งวันที่ 6 สิงหาคม จำนวน 118 ตัว, 15 สิงหาคม จำนวน 120 ตัว, และ 16 สิงหาคม จำนวน 40 ตัว ซึ่งจะได้รับสินค้าในวันที่ 13 สิงหาคม, 22 สิงหาคม, และ 23 สิงหาคมซึ่งสามารถจัดการลดสินค้าคงคลังได้จำนวน 59.09 เปอร์เซนต์

5.2 การเปรียบเทียบค่าพื้นที่ใต้กราฟ จากพื้นที่ใต้กราฟของข้อมูลจริงมีค่าพื้นที่ใต้กราฟเท่ากับ 6,335.5 หน่วย\*วัน และข้อมูลจากโปรแกรมมีค่าพื้นที่ใต้กราฟเท่ากับ 3,115 หน่วย\*วัน ซึ่งผลแตกต่างเท่ากับ 3,220.5 หน่วย\*วัน

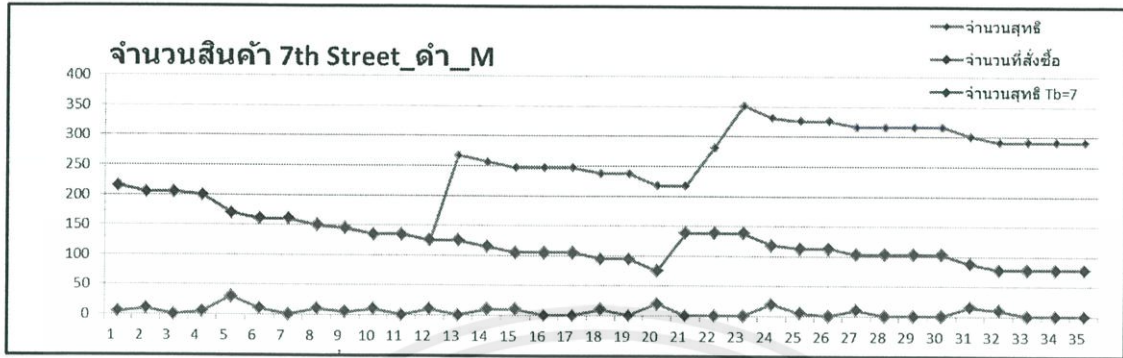
6. การปรับปรุงประสิทธิภาพสินค้ารุ่น 7th Street ต่ำ M จำนวนความต้องการสินค้าต่อวัน และจำนวนสินค้าคงคลังสุทธิจากผู้ประกอบการกับโปรแกรม

ตารางที่ 4.6 สินค้ารุ่น 7th Street ต่ำ M

เริ่มคลังที่ 220	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
7th Street ต่ำ M	1	215	215	5
7th Street ต่ำ M	2	205	205	10
7th Street ต่ำ M	3	205	205	0
7th Street ต่ำ M	4	200	200	5
7th Street ต่ำ M	5	170	170	30
7th Street ต่ำ M	6	160	160	10
7th Street ต่ำ M	7	160	160	0
7th Street ต่ำ M	8	150	150	10
7th Street ต่ำ M	9	145	145	5
7th Street ต่ำ M	10	135	135	10
7th Street ต่ำ M	11	135	135	0
7th Street ต่ำ M	12	125	125	10
7th Street ต่ำ M	13	267	125	0
7th Street ต่ำ M	14	257	115	10
7th Street ต่ำ M	15	247	105	10
7th Street ต่ำ M	16	247	105	0
7th Street ต่ำ M	17	247	105	0
7th Street ต่ำ M	18	237	95	10
7th Street ต่ำ M	19	237	95	0

ตารางที่ 4.6 สินค้ารุ่น 7th Street ต่ำ M (ต่อ)

เริ่มคงคลังที่ 220	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
7th Street_ต่ำ_M	20	217	75	20
7th Street_ต่ำ_M	21	217	138	0
7th Street_ต่ำ_M	22	281	138	0
7th Street_ต่ำ_M	23	351	138	0
7th Street_ต่ำ_M	24	331	118	20
7th Street_ต่ำ_M	25	326	113	5
7th Street_ต่ำ_M	26	326	113	0
7th Street_ต่ำ_M	27	316	103	10
7th Street_ต่ำ_M	28	316	103	0
7th Street_ต่ำ_M	29	316	103	0
7th Street_ต่ำ_M	30	316	103	0
7th Street_ต่ำ_M	31	301	88	15
7th Street_ต่ำ_M	32	291	78	10



รูปที่ 4.24 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้านำ 7th Street ดำ M

จากตารางที่ 4.6 และรูปที่ 4.24 แสดงการเปรียบเทียบยอดของสินค้าคงคลังว่ามีจำนวนสินค้าคงคลังเหลือปริมาณเท่าใดภายใต้เงื่อนไข reorder point และ วิเคราะห์ระบบสินค้าคงคลังด้วยตัวแบบสินค้าคงคลังเชิงสโตคาสติก มีดังนี้

6.1 การเปรียบเทียบระหว่างวิธีการใช้ข้อมูลจริงกับข้อมูลจากโปรแกรมเพื่อดูผลของสินค้าที่อยู่ในคลังมากเกินไปของโมเดลสินค้า 7th Street ดำ M พบว่าหลังจากการใช้เว็บแอปพลิเคชัน การสนับสนุนการตัดสินใจแล้ว สินค้า 7th Street ดำ M ต้องมีการสั่งจำนวน 1 ครั้ง คือ วันที่ 14 สิงหาคม จำนวน 63 ตัว ซึ่งจะได้รับสินค้าในวันที่ 21 สิงหาคม โดยมีแนวโน้มที่แตกต่างจากการสั่งซื้อเดิมที่สั่งวันที่ 6 สิงหาคม จำนวน 142 ตัว, 15 สิงหาคม จำนวน 64 ตัว, และ 16 สิงหาคม จำนวน 70 ตัว ซึ่งจะได้รับสินค้าในวันที่ 13 สิงหาคม, 22 สิงหาคม, และ 23 สิงหาคมซึ่งสามารถจัดการลดสินค้าคงคลังได้จำนวน 73.2 เปอร์เซ็นต์

6.2 การเปรียบเทียบค่าพื้นที่ใต้กราฟ จากพื้นที่ใต้กราฟของข้อมูลจริงมีค่าพื้นที่ใต้กราฟเท่ากับ 7,687 หน่วย\*วัน และข้อมูลจากโปรแกรมมีค่าพื้นที่ใต้กราฟเท่ากับ 4,092.5 หน่วย\*วัน ซึ่งผลแตกต่างเท่ากับ 3,594.5 หน่วย\*วัน

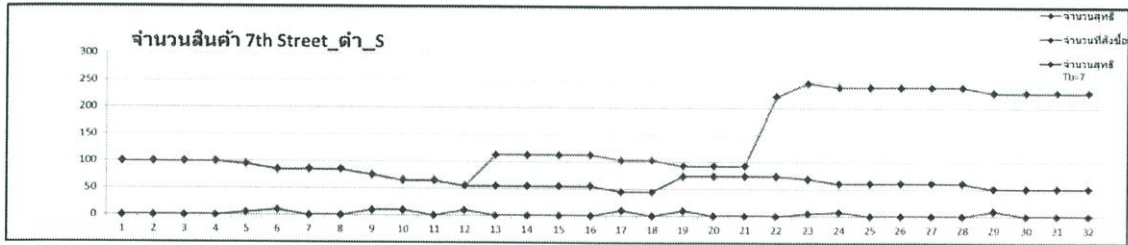
7. การปรับปรุงประสิทธิภาพสินค้ารุ่น 7th Street ดำ S จำนวนความต้องการสินค้าต่อวัน และจำนวนสินค้าคงคลังสุทธิจากผู้ประกอบการกับโปรแกรม

ตารางที่ 4.7 สินค้ารุ่น 7th Street ดำ S

เริ่มคลังที่ 100	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
7th Street ดำ_S	1	100	100	0
7th Street ดำ_S	2	100	100	0
7th Street ดำ_S	3	100	100	0
7th Street ดำ_S	4	100	100	0
7th Street ดำ_S	5	95	95	5
7th Street ดำ_S	6	85	85	10
7th Street ดำ_S	7	85	85	0
7th Street ดำ_S	8	85	85	0
7th Street ดำ_S	9	75	75	10
7th Street ดำ_S	10	65	65	10
7th Street ดำ_S	11	65	65	0
7th Street ดำ_S	12	55	55	10
7th Street ดำ_S	13	113	55	0
7th Street ดำ_S	14	113	55	0
7th Street ดำ_S	15	113	55	0
7th Street ดำ_S	16	113	55	0
7th Street ดำ_S	17	103	45	10
7th Street ดำ_S	18	103	45	0
7th Street ดำ_S	19	93	74	10

ตารางที่ 4.7 สินค้ารุ่น 7th Street ดำ S (ต่อ)

เริ่มคงคลังที่ 100	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
7th Street_ดำ_S	20	93	74	0
7th Street_ดำ_S	21	93	74	0
7th Street_ดำ_S	22	222	74	0
7th Street_ดำ_S	23	247	69	5
7th Street_ดำ_S	24	239	61	8
7th Street_ดำ_S	25	239	61	0
7th Street_ดำ_S	26	239	61	0
7th Street_ดำ_S	27	239	61	0
7th Street_ดำ_S	28	239	61	0
7th Street_ดำ_S	29	229	51	10
7th Street_ดำ_S	30	229	51	0
7th Street_ดำ_S	31	229	51	0
7th Street_ดำ_S	32	229	51	0



รูปที่ 4.25 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 7th Street ดำ S

จากตารางที่ 4.7 และรูปที่ 4.25 แสดงการเปรียบเทียบยอดของสินค้าคงคลังว่ามีจำนวนสินค้าคงคลังเหลือปริมาณเท่าใดภายใต้เงื่อนไข reorder point และ วิเคราะห์ระบบสินค้าคงคลังด้วยตัวแบบสินค้าคงคลังเชิงสโตคาสติก มีดังนี้

7.1 การเปรียบเทียบระหว่างวิธีการใช้ข้อมูลจริงกับข้อมูลจากโปรแกรมเพื่อดูผลของสินค้าที่อยู่ในคลังมากเกินไปของโมเดลสินค้า 7th Street ดำ S พบว่าหลังจากการใช้เว็บแอปพลิเคชัน การสนับสนุนการตัดสินใจแล้ว สินค้า 7th Street ดำ S ต้องมีการสั่งจำนวน 1 ครั้ง คือ วันที่ 12 สิงหาคม จำนวน 29 ตัว ซึ่งจะได้รับสินค้าในวันที่ 19 สิงหาคม โดยมีแนวโน้มที่แตกต่างจากการสั่งซื้อเดิมที่สั่งวันที่ 6 สิงหาคม จำนวน 58 ตัว, 15 สิงหาคม จำนวน 129 ตัว, และ 16 สิงหาคม จำนวน 25 ตัว ซึ่งจะได้รับสินค้าในวันที่ 13 สิงหาคม, 22 สิงหาคม, และ 23 สิงหาคมซึ่งสามารถจัดการลดสินค้าคงคลังได้จำนวน 77.73 เปอร์เซนต์

7.2 การเปรียบเทียบค่าพื้นที่ใต้กราฟ จากพื้นที่ใต้กราฟของข้อมูลจริงมีค่าพื้นที่ใต้กราฟเท่ากับ 4,591.5 หน่วย\*วัน และข้อมูลจากโปรแกรมมีค่าพื้นที่ใต้กราฟเท่ากับ 2,169.5 หน่วย\*วัน ซึ่งผลแตกต่างเท่ากับ 2,422 หน่วย\*วัน

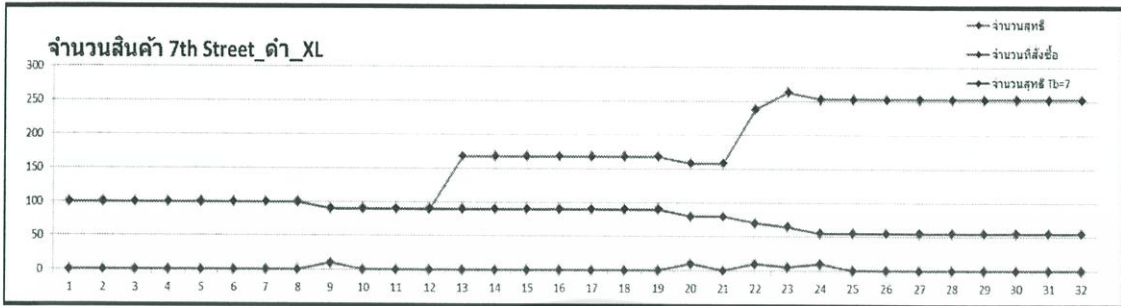
8. การปรับปรุงประสิทธิภาพสินค้ารุ่น 7th Street ดำ XL จำนวนความต้องการสินค้าต่อวัน และจำนวนสินค้าคงคลังสุทธิจากผู้ประกอบการกับโปรแกรม

ตารางที่ 4.8 สินค้ารุ่น 7th Street ดำ XL

เริ่มคงคลังที่ 100	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
7th Street_ดำ_XL	1	100	100	0
7th Street_ดำ_XL	2	100	100	0
7th Street_ดำ_XL	3	100	100	0
7th Street_ดำ_XL	4	100	100	0
7th Street_ดำ_XL	5	100	100	0
7th Street_ดำ_XL	6	100	100	0
7th Street_ดำ_XL	7	100	100	0
7th Street_ดำ_XL	8	100	100	0
7th Street_ดำ_XL	9	90	90	10
7th Street_ดำ_XL	10	90	90	0
7th Street_ดำ_XL	11	90	90	0
7th Street_ดำ_XL	12	90	90	0
7th Street_ดำ_XL	13	168	90	0
7th Street_ดำ_XL	14	168	90	0
7th Street_ดำ_XL	15	168	90	0
7th Street_ดำ_XL	16	168	90	0
7th Street_ดำ_XL	17	168	90	0
7th Street_ดำ_XL	18	168	90	0
7th Street_ดำ_XL	19	168	90	0

ตารางที่ 4.8 สินค้ารุ่น 7th Street ต่ำ XL (ต่อ)

เริ่มคงคลังที่ 100	วันที่	จำนวนสุทธิ	จำนวนสุทธิ Tb=7	จำนวนที่ สั่งซื้อ
7th Street_ต่ำ_XL	20	158	80	10
7th Street_ต่ำ_XL	21	158	80	0
7th Street_ต่ำ_XL	22	238	70	10
7th Street_ต่ำ_XL	23	263	65	5
7th Street_ต่ำ_XL	24	253	55	10
7th Street_ต่ำ_XL	25	253	55	0
7th Street_ต่ำ_XL	26	253	55	0
7th Street_ต่ำ_XL	27	253	55	0
7th Street_ต่ำ_XL	28	253	55	0
7th Street_ต่ำ_XL	29	253	55	0
7th Street_ต่ำ_XL	30	253	55	0
7th Street_ต่ำ_XL	31	253	55	0
7th Street_ต่ำ_XL	32	253	55	0



รูปที่ 4.26 การเปรียบเทียบประสิทธิภาพการทำงานของสินค้ารุ่น 7th Street ดำ XL

จากตารางที่ 4.8 และรูปที่ 4.26 แสดงการเปรียบเทียบยอดของสินค้าคงคลังว่ามีจำนวนสินค้าคงคลังเหลือปริมาณเท่าใดภายใต้เงื่อนไข reorder point และ วิเคราะห์ระบบสินค้าคงคลังด้วยตัวแบบสินค้าคงคลังเชิงสโตคาสติก มีดังนี้

8.1 การเปรียบเทียบระหว่างวิธีการใช้ข้อมูลจริงกับข้อมูลจากโปรแกรมเพื่อดูผลของสินค้าที่อยู่ในคลังมากเกินไปของโมเดลสินค้า 7th Street ดำ XL พบว่าหลังจากการใช้เว็บแอปพลิเคชัน การสนับสนุนการตัดสินใจแล้ว สินค้า 7th Street ดำ XL ไม่ต้องมีคำสั่งซื้อ โดยมีแนวโน้มที่แตกต่างจากการสั่งซื้อเดิมที่สั่งวันที่ 6 สิงหาคม จำนวน 78 ตัว, 15 สิงหาคม จำนวน 80 ตัว, และ 16 สิงหาคม จำนวน 25 ตัว ซึ่งจะได้รับสินค้าในวันที่ 13 สิงหาคม, 22 สิงหาคม, และ 23 สิงหาคมซึ่งสามารถจัดการลดสินค้าคงคลังได้จำนวน 78.26 เปอร์เซนต์

8.2 การเปรียบเทียบค่าพื้นที่ใต้กราฟ จากพื้นที่ใต้กราฟของข้อมูลจริงมีค่าพื้นที่ใต้กราฟเท่ากับ 5,506.5 หน่วย\*วัน และข้อมูลจากโปรแกรมมีค่าพื้นที่ใต้กราฟเท่ากับ 2,557.5 หน่วย\*วัน ซึ่งผลแตกต่างเท่ากับ 2,949 หน่วย\*วัน

## บทที่ 5

### สรุปผลการดำเนินงาน

#### 5.1 สรุปผลการดำเนินงาน

ปริญญานิพนธ์นี้จัดทำขึ้นเพื่อการออกแบบระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่น ผ่านรูปแบบของเว็บแอปพลิเคชัน เพื่อตอบสนองความต้องการของผู้ประกอบการในธุรกิจเสื้อผ้าแฟชั่น ซึ่งเป็นธุรกิจที่มีการหมุนเวียนของสินค้าอยู่ตลอดเวลา ทำให้เกิดความซับซ้อนในการวางแผนการจัดการสินค้าคงคลัง โดยในการจัดทำนี้มีจุดมุ่งหมายที่สำคัญคือ ทำการออกแบบและพัฒนาโปรแกรมฐานข้อมูลในการสนับสนุนการตัดสินใจการจัดการสินค้าคงคลังในธุรกิจ เสื้อผ้าแฟชั่น เพื่อสามารถนำโปรแกรมสนับสนุนการตัดสินใจไปใช้ในการบริหารจัดการสินค้าคงคลังในธุรกิจเสื้อผ้าแฟชั่นได้อย่างมีประสิทธิภาพในแนวคิดการรวมข้อมูลเข้าสู่ศูนย์กลางผ่านระบบอินเทอร์เน็ต ภายใต้อุปกรณ์การจัดจำหน่ายสินค้าจริง ของ บริษัท 7th Street ในเวลา 1 เดือน โดยการคัดเลือกสินค้าสามรุ่น รุ่นละสี่ขนาด รวมทั้งสิ้นเป็นจำนวน 12 โมเดล โดยได้ทำการใช้ข้อมูลเหล่านั้นมาป้อนเข้าสู่เว็บแอปพลิเคชัน เพื่อทดสอบการทำงานของระบบ ว่าสามารถใช้งานได้จริงและแสดงผลในส่วนที่เป็นประโยชน์ต่อการจัดการสินค้าภายในคลังได้ ประกอบไปด้วย อัตราหมุนเวียนสินค้า จำนวนที่สินค้ามีเพียงพอต่อการขาย ช่วงเวลาที่ควรทำการสั่งซื้อและปริมาณการสั่งซื้อสินค้าเข้าสู่คลัง

ผลที่ได้จากโครงการ คือ ระบบสามารถรับข้อมูลการซื้อและจำหน่ายสินค้าผ่านหน้าจอในส่วนของการบันทึกข้อมูล แล้วสามารถนำข้อมูลเหล่านั้นไปใช้ประโยชน์ในส่วนต่างๆต่อไปได้ ดังนี้ ระบบแสดงสินค้าคงคลัง ซึ่งจะแสดงปริมาณสินค้าที่มีอยู่ในปัจจุบันว่าสินค้าแต่ละประเภท แต่ละขนาดนั้นมีสินค้าอยู่ในคลังเป็นจำนวนเท่าใด โดยแสดงในรูปของตารางที่มีการแบ่งที่ง่ายต่อความเข้าใจต่อผู้ใช้งาน ต่อมาคือการนำข้อมูลการซื้อและจัดจำหน่ายไปใช้ในการแสดงผลส่วนที่ช่วยสนับสนุนในการตัดสินใจการจัดการสินค้าคงคลังของผู้ประกอบการ โดยจะแสดงสถานะสินค้า ว่ามีความต้องการเติมสินค้าหรือไม่ แบ่งออกเป็นสี่สถานะคือ สถานะสินค้าเพียงพอ เป็นสถานะที่บอกถึงว่าปัจจุบันมีสินค้าเพียงพอต่อการขาย สถานะเผื่อสำรอง คือสถานะที่สินค้าใกล้จะถึงจุดที่มีความต้องการเติมสินค้าแล้ว สถานะสั่งซื้อสินค้า คือสถานะที่ต้องทำการเติมสินค้าเพื่อตอบสนองต่อการขายในอนาคต และสถานะสุดท้าย สถานะสั่งซื้อสินค้าทันทีและเร่งการผลิต สถานะนี้จำเป็นต้องเร่งการผลิตไปยังผู้จัดส่งเพราะว่าสินค้าเกิดการขาดแคลนฉับพลัน อาจจะมีผลทำให้สินค้าขาดช่วง

หากไม่ทำการเร่งการผลิตแก่ผู้จัดส่ง ในส่วนของข้อมูลสนับสนุนการจัดการสินค้าคงคลังยังประกอบไปด้วย ข้อมูลปริมาณการสั่งซื้อที่เหมาะสม ซึ่งเมื่อเราจะต้องทำการสั่งซื้อสินค้าเราควรที่จะสั่งซื้อสินค้าในปริมาณเท่าใด และส่วนสุดท้ายประกอบไปด้วยสองส่วน คือ สามารถนำข้อมูลการจำหน่ายสินค้า มาใช้วิเคราะห์กระแสของ สินค้าว่าสินค้านั้นมีการหมุนเวียนของสินค้าที่ดีหรือไม่ โดยแสดงในส่วนของอัตราการหมุนเวียนสินค้า ซึ่งใน ส่วนนี้ยังสามารถเปรียบเทียบกับสินค้าอื่นได้อีกด้วย แต่เนื่องด้วยข้อจำกัดทางด้านเวลาและความสามารถ ทางด้านโปรแกรมทำให้ยังไม่สามารถทำระบบทั้งหมดให้นำไปใช้บนเซิร์ฟเวอร์อินเทอร์เน็ตได้จริง และบางส่วนของโปรแกรมสามารถพัฒนาเพื่อความสะดวกได้มากกว่านี้

## 5.2 ข้อเสนอแนะ

ทางคณะผู้จัดทำมีแนวทางที่จะพัฒนาระบบสนับสนุนการตัดสินใจในการจัดการสินค้าคงคลังในธุรกิจ เสื้อผ้าแฟชั่นดังนี้

- 1.พัฒนาให้ระบบทำงานผ่านระบบอินเทอร์เน็ตได้จริง
- 2.ทำให้ระบบมีการรับข้อมูลในรูปแบบของบาร์โค้ดเพื่อสะดวกต่อการป้อนข้อมูล
- 3.พัฒนาส่วนของการใช้งานให้มีความสะดวกและง่ายต่อความเข้าใจมากยิ่งขึ้น
- 4.เพิ่มความยืดหยุ่นของการใช้งาน เพื่อประยุกต์ใช้ในธุรกิจประเภทอื่นๆ

## เอกสารอ้างอิง

- [1] ความหมายของ Web Application : <http://plainsofeternity.blogspot.com/2013/07/3-web-application.html>
- [2] ศรีไพร ศักดิ์รุ่งพงศากุล, เจษฎาพร ยุทธนวิบูลย์ชัย, 2549. ระบบสารสนเทศและเทคโนโลยีการจัดการความรู้. กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- [3] โอภาส เอี่ยมศิริวงศ์, 2545. การวิเคราะห์และออกแบบระบบ. กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- [4] บุชบา พุกษาพันธุ์รัตน์, 2552. การวางแผนและควบคุมการผลิต. กรุงเทพฯ : ท้อป.
- [5] ชุมพล ศฤงคารศิริ, 2545. การวางแผนและควบคุมการผลิต ฉบับปรับปรุงใหม่. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).
- [6] นันทนี แสงโสภา, 2555. Advanced Access ฉบับเขียนโปรแกรม VBA. กรุงเทพฯ : โปริชั่น.
- [7] NetBeans IDE : [https://www.itgenius.co.th/article/โปรแกรม NetBeans.html](https://www.itgenius.co.th/article/โปรแกรม%20NetBeans.html).



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Source Code

### NETBEANS Code

#### 1. Circulation Facade

```
package demo.common.facade;

import demo.common.db.Orderdetail;

import java.text.SimpleDateFormat;

import java.util.List;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

@Stateless

public class CirculationFacade extends AbstractFacade<Orderdetail> {

    @PersistenceContext(unitName = "DemoWebAppPU")

    private EntityManager em;

    @Override

    protected EntityManager getEntityManager() {

        return em;

    }

    public CirculationFacade() {

        super(Orderdetail.class);

    }

}
```

```

}

public List<Orderdetail> searchOrderCode(String keyword) {

    Query searchQuery;

    searchQuery = em.createQuery("SELECT o FROM Orderdetail o WHERE
o.ordercode.ordercode LIKE :keyword");

    searchQuery.setParameter("keyword", "%" + keyword + "%");

    List<Orderdetail> circulationList = searchQuery.getResultList();

    return circulationList;
}

public List<Orderdetail> searchOrderCode2(String keyword) {

    Query searchQuery;

    searchQuery = em.createQuery("SELECT o FROM Orderdetail o WHERE
o.ordercode.ordercode = :keyword");

    searchQuery.setParameter("keyword", keyword);

    List<Orderdetail> circulationList = searchQuery.getResultList();

    return circulationList;
}

public List<Object[]> searchCirculationManagement(String name) {

    Query searchQuery;

    searchQuery = em.createQuery("SELECT o.ordercode.dateorder,"
+ "o.ordercode.ordercode,"

```

```

+ "o.ordercode.custCode.custCode,"
+ "o.modelcode.name,"
+ " SUM(CASE WHEN('XS' = o.modelcode.size) THEN o.number ELSE 0 END),"
+ " SUM(CASE WHEN('S' = o.modelcode.size) THEN o.number ELSE 0 END),"
+ " SUM(CASE WHEN('M' = o.modelcode.size) THEN o.number ELSE 0 END),"
+ " SUM(CASE WHEN('L' = o.modelcode.size) THEN o.number ELSE 0 END),"
+ " SUM(CASE WHEN('XL' = o.modelcode.size) THEN o.number ELSE 0 END)"
+ " FROM Orderdetail o"
+ " WHERE o.modelcode.name LIKE :name"
+ " GROUP BY
o.ordercode.dateorder,o.ordercode.ordercode,o.ordercode.custCode.custCode,o.modelcode.name");

searchQuery.setParameter("name", "%" + name + "%");

List<Object[]> circulationList = searchQuery.getResultList();

for(int i=0; i<circulationList.size(); i++) {

    circulationList.get(i)[0] = new SimpleDateFormat("yyy-MM-dd").format(circulationList.get(i)[0]);

}

System.out.println("circulationList.size " + circulationList.size() + " " + name);

return circulationList;

}

```

```
}
```

## 2. Customer Façade

```
package demo.common.facade;
```

```
import demo.common.db.Customer;
```

```
import java.util.List;
```

```
import javax.ejb.Stateless;
```

```
import javax.persistence.EntityManager;
```

```
import javax.persistence.PersistenceContext;
```

```
import javax.persistence.Query;
```

```
@Stateless
```

```
public class CustomerFacade extends AbstractFacade<Customer> {
```

```
    @PersistenceContext(unitName = "DemoWebAppPU")
```

```
    private EntityManager em;
```

```
    @Override
```

```
    protected EntityManager getEntityManager() {
```

```
        return em;
```

```
    }
```

```
    public CustomerFacade() {
```

```
        super(Customer.class);
```

```
    }
```

```

public List<Customer> searchCustomer(String keyword) {

    Query searchQuery;

    searchQuery = em.createQuery("SELECT c FROM Customer c WHERE c.custCode =
:keyword");

    searchQuery.setParameter("keyword", keyword);

    List<Customer> customerList = searchQuery.getResultList();

    return customerList;

}
}

```

Decision Façade

```

package demo.common.facade;

import static com.sun.org.apache.xalan.internal.xsltc.compiler.util.Type.Int;

import demo.circulation.ctrl.CirculationController;

import demo.common.db.Orderdetail;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;

```

```

import javax.persistence.PersistenceContext;

import javax.persistence.Query;

@Stateless

public class DecisionFacade extends AbstractFacade<Orderdetail> {

    @PersistenceContext(unitName = "DemoWebAppPU")

    private EntityManager em;

    @Override

    protected EntityManager getEntityManager() {

        return em;

    }

    public DecisionFacade() {

        super(Orderdetail.class);

    }

    public List<Orderdetail> searchDecision(Date keyword) {

        Query searchQuery;

        searchQuery = em.createQuery("SELECT d FROM Orderdetail d WHERE d.dateorder =

keyword");

        searchQuery.setParameter("keyword", "%" + keyword + "%");

        List<Orderdetail> decisionList = searchQuery.getResultList();

        return decisionList;

    }
}

```

```

public List<Object[]> searchDecisionManagement(Date keyword) {

    int numberLookbackDay = 10;

    int tb = 7;

    int L = 7;

    double z = 1.65;

    Query searchQuery;

    searchQuery = em.createQuery("SELECT p.modelcode.modelcode, p.modelcode.name,
p.modelcode.size,"
        + " SUM(p.number),"
        + " SUM(p.number),"
        + " SUM(p.number)"
        + " FROM Product p"
        + " WHERE p.createdDate <= :enddate"
        + " GROUP BY p.modelcode.modelcode, p.modelcode.name, p.modelcode.size");

    searchQuery.setParameter("enddate", keyword);

    List<Object[]> modelList = searchQuery.getResultList();

    for(int i=0; i<modelList.size(); i++) {

        System.out.println("ModelCode :>> " + modelList.get(i)[0]);

        searchQuery = em.createQuery("SELECT p.modelcode.modelcode, p.modelcode.name,
p.modelcode.size, p.ordercode.dateorder,"
            + " SUM(p.number)"

```

```

+ " FROM Orderdetail p"

+ " WHERE p.ordercode.dateorder BETWEEN :startdate AND :enddate"

+ " AND p.modelcode.modelcode = :modelcode"

+ " GROUP BY p.modelcode.modelcode, p.modelcode.name,
p.modelcode.size,p.ordercode.dateorder"

+ " ORDER BY p.ordercode.dateorder DESC");

Calendar cal = Calendar.getInstance();
//Date dateInstance = new Date();
cal.setTime(keyword);
System.out.println("START DATE ::::::::::: " + cal.getTime().toString());
cal.add(Calendar.DATE, -1*(numberLookbackDay));
Date startDate = cal.getTime();
System.out.println("END DATE ::::::::::: " + cal.getTime().toString());
searchQuery.setParameter("startdate", startDate);
searchQuery.setParameter("enddate", keyword);
searchQuery.setParameter("modelcode", modelList.get(i)[0]);
//searchQuery.setMaxResults(1);

List<Object[]> itemSoldList = searchQuery.getResultList();

searchQuery = em.createQuery("SELECT p.modelcode.modelcode, p.modelcode.name,
p.modelcode.size,"
+ " SUM(p.number)"

```

```

+ " FROM Orderdetail p"

+ " WHERE p.ordercode.dateorder <= :enddate"

+ " AND p.modelcode.modelcode = :modelcode"

+ " GROUP BY p.modelcode.modelcode, p.modelcode.name, p.modelcode.size");

searchQuery.setParameter("enddate", keyword);

searchQuery.setParameter("modelcode", modelList.get(i)[0]);

List<Object[]> sumOrderList = searchQuery.getResultList();

if(!sumOrderList.isEmpty()) {

    modelList.get(i)[4] = (long)modelList.get(i)[4] - (long)sumOrderList.get(0)[3];

}

if(itemSoldList.isEmpty()) {

    System.out.println("No order");

    modelList.get(i)[3] = "สินค้าเพียงพอ";

    //System.out.println("[avg:]" + (0*(tb + L) + ((z*0*Math.sqrt((tb + L)))))+ " [st:]" +
(z*0*Math.sqrt((tb + L))));

    modelList.get(i)[5] = 0;

} else {

    double sum = 0;

    for(int j=0; j<itemSoldList.size(); j++){

        sum = sum + (long)itemSoldList.get(j)[4];

    }

```

```

double avg = (double)sum/numberLookbackDay;

sum = 0;

for(int j=0; j<itemSoldList.size(); j++){

    sum = sum + Math.pow((long)itemSoldList.get(j)[4] - avg, 2);

}

double std = Math.sqrt((sum+(avg*avg*(numberLookbackDay-
itemSoldList.size())))/(numberLookbackDay-1));

double R = Math.round((avg*L) + (z*std*Math.sqrt(L))*1)/1.0;

int Q = (int) Math.round((avg*(tb + L) + (z*std*Math.sqrt((tb + L))) -
((long)modelList.get(i)[4]));

System.out.println("[avg:]" + avg + " [std:]" + std + " [count:]" + itemSoldList.size());

System.out.println("sum :>>>> " + sum);

System.out.println("Reorder Point :>>>> " + R);

System.out.println("I  :>>>> " + modelList.get(i)[4]);

System.out.println("[avg:]" + (avg*(tb + L) + ((z*std*Math.sqrt((tb + L)))) + " [st:]" +
(z*std*Math.sqrt((tb + L))));

if((R <= (long)modelList.get(i)[4]) && ((long)modelList.get(i)[4] < 1.2*R))

    modelList.get(i)[3] = "สั่งซื้อทันที!!!!" ;

else if((1.2*R < (long)modelList.get(i)[4]) && ((long)modelList.get(i)[4] < 1.5*R))

    modelList.get(i)[3] = "เฝ้าระวัง" ;

else if(1.5*R < (long)modelList.get(i)[4])

```

```

        modelList.get(i)[3] = "สินค้าเพียงพอ" ;

else if(R > (long)modelList.get(i)[4])

        modelList.get(i)[3] = "สั่งซื้อทันที !! และเร่งการผลิต" ;

else if ((long)modelList.get(i)[4] == 0 )

        modelList.get(i)[3] = "สินค้าหมด" ;

else

        modelList.get(i)[3] = "สั่งซื้อทันที !! และเร่งการผลิต" ;

//modelList.get(i)[3] = R;

if(1.5*R < (long)modelList.get(i)[4])

        modelList.get(i)[5] = 0;

else if(1.2*R > (long)modelList.get(i)[4])

        modelList.get(i)[5] = Q;

else

        modelList.get(i)[5] = 0;

}

}

return modelList;

}

}

```

### 3. Inventory Façade

```
package demo.common.facade;

import demo.common.db.Product;

import java.util.List;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

@Stateless
public class InventoryFacade extends AbstractFacade<Product> {

    @PersistenceContext(unitName = "DemoWebAppPU")
    private EntityManager em;

    @Override
    protected EntityManager getEntityManager() {

        return em;
    }

    public InventoryFacade() {

        super(Product.class);
    }

    public List<Product> searchInventory(String keyword) {

        Query searchQuery;
```

```
searchQuery = em.createQuery("SELECT p FROM Product p WHERE p.itemid LIKE  
:keyword");
```

```
searchQuery.setParameter("keyword", "%" + keyword + "%");
```

```
List<Product> inventoryList = searchQuery.getResultList();
```

```
return inventoryList;
```

```
}
```

```
public List<Object[]> searchInventoryManagement(String name) {
```

```
Query searchQuery;
```

```
searchQuery = em.createQuery("SELECT p.modelcode.name,"
```

```
+ " SUM(CASE WHEN('XS' = p.modelcode.size) THEN p.number ELSE 0 END),"
```

```
+ " SUM(CASE WHEN('S' = p.modelcode.size) THEN p.number ELSE 0 END),"
```

```
+ " SUM(CASE WHEN('M' = p.modelcode.size) THEN p.number ELSE 0 END),"
```

```
+ " SUM(CASE WHEN('L' = p.modelcode.size) THEN p.number ELSE 0 END),"
```

```
+ " SUM(CASE WHEN('XL' = p.modelcode.size) THEN p.number ELSE 0 END)"
```

```
+ " FROM Product p"
```

```
+ " WHERE p.modelcode.name LIKE :name"
```

```
+ " GROUP BY p.modelcode.name");
```

```
searchQuery.setParameter("name", "%" + name + "%");
```

```
List<Object[]> inventoryList = searchQuery.getResultList();
```

```
searchQuery = em.createQuery("SELECT p.modelcode.name,"
```

```
+ " SUM(CASE WHEN('XS' = p.modelcode.size) THEN p.number ELSE 0 END),"
```

```

+ " SUM(CASE WHEN('S' = p.modelcode.size) THEN p.number ELSE 0 END),"
+ " SUM(CASE WHEN('M' = p.modelcode.size) THEN p.number ELSE 0 END),"
+ " SUM(CASE WHEN('L' = p.modelcode.size) THEN p.number ELSE 0 END),"
+ " SUM(CASE WHEN('XL' = p.modelcode.size) THEN p.number ELSE 0 END)"
+ " FROM Orderdetail p"
+ " WHERE p.modelcode.name LIKE :name"
+ " GROUP BY p.modelcode.name");
searchQuery.setParameter("name", "%" + name + "%");
List<Object[]> templInventoryList = searchQuery.getResultList();
for(int i=0; i<inventoryList.size(); i++) {
    for(int j=0; j<templInventoryList.size(); j++) {
        if(inventoryList.get(i)[0].equals(templInventoryList.get(j)[0])) {
            inventoryList.get(i)[1] = (int) inventoryList.get(i)[1] - (int) templInventoryList.get(j)[1];
            inventoryList.get(i)[2] = (int) inventoryList.get(i)[2] - (int) templInventoryList.get(j)[2];
            inventoryList.get(i)[3] = (int) inventoryList.get(i)[3] - (int) templInventoryList.get(j)[3];
            inventoryList.get(i)[4] = (int) inventoryList.get(i)[4] - (int) templInventoryList.get(j)[4];
            inventoryList.get(i)[5] = (int) inventoryList.get(i)[5] - (int) templInventoryList.get(j)[5];
            templInventoryList.remove(j);
            break;
        }
    }
}

```

```

    }

    }

    return inventoryList;

}
}

```

#### 4. Inventoryturn Facade

```

package demo.common.facade;

import demo.common.db.Product;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import javax.persistence.Query;

@Stateless

public class InventoryturnFacade extends AbstractFacade<Product> {

    @PersistenceContext(unitName = "DemoWebAppPU")

    private EntityManager em;

    @Override

    protected EntityManager getEntityManager() {

```

```

return em;
}

public InventoryturnFacade() {
    super(Product.class);
}

public List<Product> searchInventory(String keyword) {
    Query searchQuery;
    searchQuery = em.createQuery("SELECT p FROM Product p WHERE p.itemid LIKE
:keyword");
    searchQuery.setParameter("keyword", "%" + keyword + "%");
    List<Product> inventoryturnList = searchQuery.getResultList();
    return inventoryturnList;
}

public List<Object[]> searchInventoryturnManagement() {
    int A = 100;
    List<Object[]> resultList = null;
    Query searchQuery;
    searchQuery = em.createQuery("SELECT DISTINCT(o.modelcode.modelcode)"
        + " FROM Product o");
    List<Object[]> modelList = searchQuery.getResultList();
    for(int i=0; i<modelList.size(); i++) {

```

```

System.out.println("ModelCode :>> " + modelList.get(i));

searchQuery = em.createQuery("SELECT p.modelcode.modelcode,"

    + " p.createdDate"

    + " FROM Product p"

    + " WHERE p.modelcode.modelcode = :modelcode"

    + " ORDER BY p.createdDate DESC");

searchQuery.setParameter("modelcode", modelList.get(i));

searchQuery.setMaxResults(1);

List<Object[]> productList = searchQuery.getResultList();

searchQuery = em.createQuery("SELECT p.modelcode.modelcode, p.modelcode.name,

p.modelcode.size,"

    + " SUM(p.number)"

    + " FROM Orderdetail p"

    + " WHERE p.ordercode.dateorder BETWEEN :startdate AND CURRENT_DATE"

    + " AND p.modelcode.modelcode = :modelcode"

    + " GROUP BY p.modelcode.modelcode, p.modelcode.name, p.modelcode.size");

searchQuery.setParameter("modelcode", modelList.get(i));

searchQuery.setParameter("startdate", productList.get(0)[1]);

List<Object[]> sumOrderList = searchQuery.getResultList();

Date currentDate = new Date();

Date startDate = (Date)productList.get(0)[1];

```

```

long diff = currentDate.getTime() - startDate.getTime();

long days = diff/(24*60*60*1000);

System.out.println("DATE :::: " + currentDate.toString() + " " + startDate.toString());

System.out.println("DATE DIFF :::::::::: " + days);

searchQuery = em.createQuery("SELECT p.modelcode.modelcode, p.modelcode.name,
p.modelcode.size,"

+ " SUM(p.number)"

+ " FROM Orderdetail p"

+ " WHERE p.ordercode.dateorder < :enddate"

+ " AND p.modelcode.modelcode = :modelcode"

+ " GROUP BY p.modelcode.modelcode, p.modelcode.name, p.modelcode.size");

searchQuery.setParameter("modelcode", modelList.get(i));
searchQuery.setParameter("enddate", productList.get(0)[1]);
List<Object[]> sumOrderList2 = searchQuery.getResultList();

long numberOrderBefore;

if(sumOrderList2.isEmpty()) {

    numberOrderBefore = 0;

} else {

    numberOrderBefore = (long)sumOrderList2.get(0)[3];

}

```

```

searchQuery = em.createQuery("SELECT p.modelcode.modelcode, p.modelcode.name,
p.modelcode.size,"

+ " SUM(p.number),"

+ " SUM(p.number)"

+ " FROM Product p"

+ " WHERE p.createdDate <= :enddate"

+ " AND p.modelcode.modelcode = :modelcode"

+ " GROUP BY p.modelcode.modelcode, p.modelcode.name, p.modelcode.size");
searchQuery.setParameter("modelcode", modelList.get(i));
searchQuery.setParameter("enddate", productList.get(0)[1]);
if(resultList == null) {
    resultList = searchQuery.getResultList();
    if(sumOrderList.isEmpty()) {
        resultList.get(i)[3] = 0;
    } else {
        System.out.println(i + " [1:]" + sumOrderList.get(0)[3] + " [2:]" + resultList.get(i)[3] +
" [3:]" + numberOrderBefore);

        resultList.get(i)[3] =
Math.round((((long)sumOrderList.get(0)[3]*A)/((((long)resultList.get(i)[3] - numberOrderBefore) +
((long)resultList.get(i)[3] - numberOrderBefore -
(long)sumOrderList.get(0)[3]))/2.0))/1.0*100.0)/100.0;
    }
}

```

```

////////////////////////////////////// TO DO

if(sumOrderList.isEmpty()) {

    resultList.get(i)[4] = 0;

} else {

    System.out.println(i + " [1]:" + sumOrderList.get(0)[3] + " [2]:" + resultList.get(i)[4] +
" [3]:" + numberOrderBefore);

    resultList.get(i)[4] = Math.round(((((((long)resultList.get(i)[4] - numberOrderBefore)
+ ((long)resultList.get(i)[4] - numberOrderBefore -
(long)sumOrderList.get(0)[3]))/2.0))/(((long)sumOrderList.get(0)[3]*1))*days)*100.0)/100.0;

}

} else {

List<Object[]> curentSellList = searchQuery.getResultList();

if(sumOrderList.isEmpty()) {

    curentSellList.get(0)[3] = 0;

} else {

    System.out.println(i + " [1]:" + sumOrderList.get(0)[3] + " [2]:" +
curentSellList.get(0)[3] + " [3]:" + numberOrderBefore);

    curentSellList.get(0)[3] =
Math.round(((((((long)sumOrderList.get(0)[3]*A)/((((long)curentSellList.get(0)[3] -
numberOrderBefore) + ((long)curentSellList.get(0)[3] - numberOrderBefore -
(long)sumOrderList.get(0)[3]))/2.0))/1.0)*100.0)/100.0;

}

////////////////////////////////////// TO DO

```

```

if(sumOrderList.isEmpty()) {

    curentSellList.get(0)[4] = 0;

} else {

    System.out.println(i + " [1]:" + sumOrderList.get(0)[3] + " [2]:" +
curentSellList.get(0)[4] + " [3]:" + numberOrderBefore);

    curentSellList.get(0)[4] = Math.round(((((((long)curentSellList.get(0)[4] -
numberOrderBefore) + ((long)curentSellList.get(0)[4] - numberOrderBefore -
(long)sumOrderList.get(0)[3]))/2.0))/(((long)sumOrderList.get(0)[3]*1))*days)*100.0)/100.0;

}

resultList.addAll(curentSellList);

}

}

return resultList;

}

}

```

## 5. Model Façade

```

package demo.common.facade;

import demo.common.db.Model;

import java.util.List;

import javax.ejb.Stateless;

```

```

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import javax.persistence.Query;

@Stateless

public class ModelFacade extends AbstractFacade<Model> {

    @PersistenceContext(unitName = "DemoWebAppPU")

    private EntityManager em;

    @Override

    protected EntityManager getEntityManager() {

        return em;

    }

    public ModelFacade() {

        super(Model.class);

    }

    public List<Model> searchModel(String keyword) {

        Query searchQuery;

        searchQuery = em.createQuery("SELECT m FROM Model m WHERE m.modelcode LIKE

:keyword");

        searchQuery.setParameter("keyword", "%" + keyword + "%");

        List<Model> modelList = searchQuery.getResultList();

        return modelList;

    }
}

```

```

}

public List<Object[]> searchModelManagement(String name) {

    Query searchQuery;

    searchQuery = em.createQuery("SELECT m.modelcode, m.name, m.size,"

        + "m.type "

        + " FROM Model m"

        + " WHERE m.name LIKE :name"

        + " GROUP BY m.modelcode,m.name,m.size,m.type"

        + " ORDER BY m.modelcode ASC");

    searchQuery.setParameter("name", "%" + name + "%");

    List<Object[]> modelList = searchQuery.getResultList();

    return modelList;

}

}

```

## 6. Order Facade

```

package demo.common.facade;

import demo.common.db.Order;

import java.util.List;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

```

```

import javax.persistence.Query;

@Stateless

public class OrderFacade extends AbstractFacade<Order> {

    @PersistenceContext(unitName = "DemoWebAppPU")

    private EntityManager em;

    @Override

    protected EntityManager getEntityManager() {

        return em;

    }

    public OrderFacade() {

        super(Order.class);

    }

    public List<Order> searchOrder(String keyword) {

        Query searchQuery;

        searchQuery = em.createQuery("SELECT o FROM Order o WHERE o.ordercode = :keyword");

        searchQuery.setParameter("keyword", keyword);

        List<Order> orderList = searchQuery.getResultList();

        return orderList;

    }

}

```

## 7. Purchase Facade

```
package demo.common.facade;

import demo.common.db.Product;

import java.text.SimpleDateFormat;

import java.util.List;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

@Stateless

public class PurchaseFacade extends AbstractFacade<Product> {

    @PersistenceContext(unitName = "DemoWebAppPU")

    private EntityManager em;

    @Override

    protected EntityManager getEntityManager() {

        return em;

    }

    public PurchaseFacade() {

        super(Product.class);

    }

}
```

```

public List<Product> searchPurchase(String keyword) {

    Query searchQuery;

    searchQuery = em.createQuery("SELECT p FROM Purchaseorder p WHERE p.purchaseid
LIKE :keyword");

    searchQuery.setParameter("keyword", "%" + keyword + "%");

    List<Product> purchaseList = searchQuery.getResultList();

    return purchaseList;
}

public List<Object[]> searchOrderManagement(String name) {

    Query searchQuery;

    searchQuery = em.createQuery("SELECT p.createdDate,"
+ "p.supcode.supcode,"
+ "p.modelcode.name,"
+ " SUM(CASE WHEN('XS' = p.modelcode.size) THEN p.number ELSE 0 END),"
+ " SUM(CASE WHEN('S' = p.modelcode.size) THEN p.number ELSE 0 END),"
+ " SUM(CASE WHEN('M' = p.modelcode.size) THEN p.number ELSE 0 END),"
+ " SUM(CASE WHEN('L' = p.modelcode.size) THEN p.number ELSE 0 END),"
+ " SUM(CASE WHEN('XL' = p.modelcode.size) THEN p.number ELSE 0 END)"
+ " FROM Product p"
+ " WHERE p.modelcode.name LIKE :name"
+ " GROUP BY p.createdDate, p.supcode.supcode, p.modelcode.name");
}

```

```

searchQuery.setParameter("name", "%" + name + "%");

List<Object[]> purchaseList = searchQuery.getResultList();

for(int i=0; i<purchaseList.size(); i++) {

    purchaseList.get(i)[0] = new SimpleDateFormat("yyy-MM-
dd").format(purchaseList.get(i)[0]);

}

return purchaseList;

}
}

```

## 8. Supplier Facade

```

package demo.common.facade;

import demo.common.db.Supplier;

import java.util.List;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import javax.persistence.Query;

@Stateless

public class SupplierFacade extends AbstractFacade<Supplier> {

    @PersistenceContext(unitName = "DemoWebAppPU")

    private EntityManager em;

```

```

@Override

protected EntityManager getEntityManager() {

    return em;

}

public SupplierFacade() {

    super(Supplier.class);

}

public List<Supplier> searchSupplier(String keyword) {

    Query searchQuery;

    searchQuery = em.createQuery("SELECT s FROM Supplier s WHERE s.supcode LIKE
:keyword");

    searchQuery.setParameter("keyword", "%" + keyword + "%");

    List<Supplier> supplierList = searchQuery.getResultList();

    return supplierList;

}

}

```