ระบบจัดการคลังสินค้า

Stock Management

นายพลากร แพทย์นุเคราะห์	รหัส 6806022510092	Sec 2
นายวุฒิพงษ์ วังรัตน์	รหัส 6806022510106	Sec 2
นางสาวฐิติรัตน์ แดงประเสริฐ	รหัส 6806022510297	Sec 2

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2568

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

คำนำ

การจัดทำโครงงาน “ระบบจัดการคลังสินค้า” นี้เป็นส่วนหนึ่งของวิชา Computer Programming ของหลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ เพื่อให้นักศึกษาได้นำความรู้ที่เรียนมาทั้งหมดมาประยุกต์ใช้ในการพัฒนาโปรแกรมที่สามารถทำงานได้จริง โดยเน้นการออกแบบและเขียนโปรแกรมด้วยภาษา python ซึ่งเป็นภาษาที่เรียนมาในวิชา Computer Programing โดยโครงงานนี้จะช่วยการคิดวิเคราะห์และการแก้ปัญหาทางเทคนิค เพื่อเตรียมความพร้อมในการประกอบอาชีพด้านวิศวกรรมสารสนเทศและเครือข่ายในอนาคต

คณะผู้จัดทำหวังว่า รายงานฉบับนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักเรียน นักศึกษา ที่กำลังหาข้อมูลเรื่องนี้อยู่ หากมีข้อแนะนำหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้และขอ อภัยมา ณ ที่นี้ด้วย

สารบัญ

	หน้า
คำนำ	ก
สารบัญ	ข
สารบัญ (ต่อ)	ค
สารบัญรูปภาพ	ง
สารบัญรูปภาพ (ต่อ)	จ
สารบัญรูปภาพ (ต่อ)	ฉ
สารบัญรูปภาพ (ต่อ)	ช
สารบัญตาราง	ซ
บทที่ 1 บทนำ	
1.1 วัตถุประสงค์ของโครงงาน	1
1.2 ขอบเขตของโครงงาน	1
1.3 ประโยชน์ที่ได้รับ	2
1.4 เครื่องมือที่คาดว่าจะต้องใช้	2
บทที่ 2 ระบบจัดการคลังสินค้า	
2.1 แฟ้มข้อมูล Product.bin	3
2.2 แฟ้มข้อมูล Sales.bin	5
2.3 แฟ้มข้อมูล Purchases.bin	6
2.4 แฟ้มข้อมูล Report.bin	8
บทที่ 3 การใช้งานระบบจัดการคลังสินค้า	
3.1 การใช้งานโปรแกรมระบบจัดการคลังสินค้า	10
3.2 การใช้งานโปรแกรมเพิ่มข้อมูล	15
3.3 การใช้งานโปรแกรมแสดงข้อมูล	16
3.4 การใช้งานโปรแกรมแก้ไขข้อมูล	22
3.5 การใช้งานโปรแกรมลบข้อมูล	24

สารบัญ (ต่อ)

หน้า

บทที่ 4 อธิบายการทำงานของ Code

4.1 ฟังก์ชั่นไบนารีพื้นฐานในระบบจัดการคลังสินค้า 31

4.2 โครงสร้างข้อมูลหลักของโปรแกรม 32

4.3 ฟังก์ชันการทำงานเมนูหลัก 33

4.4 ฟังก์ชันเมนูระบบจัดการข้อมูลสินค้า 36

4.5 ฟังก์ชันเมนูระบบจัดการข้อมูลการขาย 51

4.6 ฟังก์ชันเมนูระบบจัดการข้อมูลการสั่งซื้อ 56

บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน 61

5.2 ปัญหาและอุปสรรคในการดำเนินงาน 61

5.3 ข้อเสนอแนะ 61

5.4 สิ่งที่ผู้จัดทำได้รับในการพัฒนาโครงงาน 62

สารบัญรูปภาพ

หน้า

รูปภาพที่ 2-1 ไฟล์ report 8
รูปภาพที่ 3-1 การเลือกใช้งานฟังก์ชั่น สินค้า 11
รูปภาพที่ 3-2 เมนูของ สินค้า 11
รูปภาพที่ 3-3 การเลือกใช้งานฟังก์ชั่น การขายสินค้า 12
รูปภาพที่ 3-4 เมนูของ การขายสินค้า 12
รูปภาพที่ 3-5 การเลือกใช้งานฟังก์ชั่น คลังสินค้า 13
รูปภาพที่ 3-6 เมนูของ คลังสินค้า 13
รูปภาพที่ 3-7 การเลือกใช้งานฟังก์ชั่น การรายงาน 14
รูปภาพที่ 3-8 การเลือกใช้งานฟังก์ชั่นของ Exit 14
รูปภาพที่ 3-9 การเลือกใช้งานฟังก์ชั่น Add product 15
รูปภาพที่ 3- 10 การเพิ่มสินค้า 15
รูปภาพที่ 3- 11 การเลือกใช้งานฟังก์ชั่น Read product 16
รูปภาพที่ 3-12 แสดงข้อมูล Read product 16
รูปภาพที่ 3-13 แสดงข้อมูล Read all product 17
รูปภาพที่ 3-14 แสดงข้อมูล สินค้าที่เลือก โดยรหัสสินค้า 17
รูปภาพที่ 3-15 การเลือกใช้งานฟังก์ชั่น Read sale details 18
รูปภาพที่ 3- 16 แสดงเมนูRead sale details 18
รูปภาพที่ 3- 17 แสดงข้อมูล Read all fields 19
รูปภาพที่ 3- 18 แสดงข้อมูล Read Id 19
รูปภาพที่ 3- 19 แสดงข้อมูล Read purchase details 20
รูปภาพที่ 3- 20 แสดงเมนูRead purchase details 20
รูปภาพที่ 3- 21 แสดงข้อมูล Read all fields 21
รูปภาพที่ 3- 22 แสดงข้อมูล Read by Id 21

สารบัญรูปภาพ (ต่อ)

หน้า

รูปภาพที่ 3- 23 การเลือกใช้งานฟังก์ชั่น Update product 22
รูปภาพที่ 3- 24 การแก้ไขข้อมูล Update product 22
รูปภาพที่ 3- 25 การเลือกใช้งานฟังก์ชั่น Update sale details by Id 23
รูปภาพที่ 3- 26 การแก้ไขข้อมูล Update sale details by Id 23
รูปภาพที่ 3- 27 การเลือกใช้งานฟังก์ชั่น Update purchase details by Id 24
รูปภาพที่ 3- 28 การแก้ไขข้อมูล Update purchare details by Id 24
รูปภาพที่ 3- 29 การเลือกใช้งานฟังก์ชั่น Delete product 25
รูปภาพที่ 3- 30 แสดงเมนู Delete product 25
รูปภาพที่ 3- 31 การลบข้อมูล Delete specific product 26
รูปภาพที่ 3- 32 การลบข้อมูล Delete all products 26
รูปภาพที่ 3- 33 การเลือกใช้งานฟังก์ชั่น Delete sale details 27
รูปภาพที่ 3- 34 แสดงเมนูDelete sale details 27
รูปภาพที่ 3- 35 แสดงเมนูDelete all sales 28
รูปภาพที่ 3- 36 แสดงเมนูDelete all details 28
รูปภาพที่ 3- 37 การเลือกใช้งานฟังก์ชั่น Delete purchase details 28
รูปภาพที่ 3- 38 แสดงเมนูDelete purchase details 29
รูปภาพที่ 3- 39 การลบ Delete purchase details 29
รูปภาพที่ 3- 40 การลบ Delete all purchases 30
รูปภาพที่ 4- 1 Code Module pickle 31
รูปภาพที่ 4- 2 Code Module datetime 31
รูปภาพที่ 4- 3 Code Module os 32
รูปภาพที่ 4- 4 โครงสร้างข้อมูลสินค้า 32
รูปภาพที่ 4- 5 โครงสร้างข้อมูลการขาย 32

สารบัญรูปภาพ (ต่อ)

หน้า

รูปภาพที่ 4- 6 โครงสร้างข้อมูลการสั่งซื้อ 32
รูปภาพที่ 4- 7 ฟังก์ชัน product_handler 33
รูปภาพที่ 4- 8 ฟังก์ชัน sale_handler 34
รูปภาพที่ 4- 9 ฟังก์ชัน purchase_handler 35
รูปภาพที่ 4- 10 การแสดงผลจากฟังก์ชัน print_report 36
รูปภาพที่ 4- 11 การกำหนดรหัสสินค้า 37
รูปภาพที่ 4- 12 การกำหนดชื่อสินค้า 37
รูปภาพที่ 4- 13 การกำหนดประเภทสินค้า 38
รูปภาพที่ 4- 14 การกำหนดหน่วยนับของสินค้า 38
รูปภาพที่ 4- 15 การกำหนดราคาขายและสถานะ 39
รูปภาพที่ 4- 16 ฟังก์ชันแสดงข้อมูลสินค้า 40
รูปภาพที่ 4- 17 การแสดงสินค้าทั้งหมด 40
รูปภาพที่ 4- 18 การแสดงสินค้าจากการค้นหาด้วยรหัสสินค้า 41
รูปภาพที่ 4- 19 ฟังก์ชันการแก้ไขข้อมูลสินค้า 42
รูปภาพที่ 4- 20 การกรอกรหัสสินค้าที่ต้องการแก้ไข 43
รูปภาพที่ 4- 21 การระบุฟิลด์ที่ต้องการแก้ไข 43
รูปภาพที่ 4- 22 การแก้ไขชื่อและการแก้ไขประเภท 44
รูปภาพที่ 4- 23 การแก้ไขจำนวนและการแก้ไขหน่วยนับ 44
รูปภาพที่ 4- 24 การแก้ไขจำนวนและการแก้ไขหน่วยนับ 45
รูปภาพที่ 4- 25 ฟังก์ชันการลบข้อมูลสินค้า 46
รูปภาพที่ 4- 26 การลบข้อมูลสินค้ารายการเดียว 46
รูปภาพที่ 4- 27 การลบข้อมูลสินค้าทั้งหมด 47
รูปภาพที่ 4- 28 ฟังก์ชันการขายสินค้าและอัปเดตสต็อก 48

สารบัญรูปภาพ (ต่อ)

หน้า

รูปภาพที่ 4- 29 การระบุสินค้าและจำนวนที่ขาย 48
รูปภาพที่ 4- 30 ฟังก์ชันการสั่งซื้อและอัปเดตสต็อก 49
รูปภาพที่ 4- 31 การระบุสินค้าและจำนวนที่สั่งซื้อ 50
รูปภาพที่ 4- 32 ฟังก์ชันเมนูระบบจัดการข้อมูลการขาย 51
รูปภาพที่ 4- 33 การแสดงข้อมูลการขายทั้งหมด 52
รูปภาพที่ 4- 34 การแสดงข้อมูลการขายเฉพาะรายการที่ต้องการค้นหา 52
รูปภาพที่ 4- 35 ฟังก์ชันการอัปเดตข้อมูลการขายตาม Sale ID 53
รูปภาพที่ 4- 36 การแก้ไขจำนวนสินค้าและยอดรวม 54
รูปภาพที่ 4- 37 ฟังก์ชันลบข้อมูลการขาย 54
รูปภาพที่ 4- 38 การลบข้อมูลการขายรายการเดียว 55
รูปภาพที่ 4- 39 การลบข้อมูลการขายทั้งหมด 55
รูปภาพที่ 4- 40 ฟังก์ชันเมนูระบบจัดการข้อมูลการสั่งซื้อ 56
รูปภาพที่ 4- 41 การแสดงข้อมูลการสั่งซื้อทั้งหมด 57
รูปภาพที่ 4- 42 การแสดงข้อมูลการสั่งซื้อรายการเดียว 57
รูปภาพที่ 4- 43 ฟังก์ชันอัปเดตข้อมูลการสั่งซื้อ 58
รูปภาพที่ 4- 44 ฟังก์ชันลบข้อมูลการสั่งซื้อ 59
รูปภาพที่ 4- 45 การลบข้อมูลการสั่งซื้อรายการเดียว 59
รูปภาพที่ 4- 46 การลบข้อมูลการสั่งซื้อทั้งหมด 60

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แฟ้มข้อมูลสินค้า	3
ตารางที่ 2.2 แฟ้มข้อมูลการขายสินค้า	5
ตารางที่ 2.3 แฟ้มข้อมูลการสั่งซื้อสินค้าเข้าสู่คลัง	6

บทที่ 1
บทนำ

1.1 วัตถุประสงค์ของโครงงาน

1.1.1 เพื่อพัฒนาระบบจัดการคลังสินค้าได้อย่างมีประสิทธิภาพ

1.1.2 เพื่อฝึกฝนทักษะการเขียนโปรแกรมด้วย Python

1.1.3 เพื่อเรียนรู้วิธีการจัดการข้อมูลและไฟล์

1.1.4 เพื่อเรียนรู้การทำงานร่วมกันเป็นทีม

1.2 ขอบเขตของโครงงาน

1.2.1 ระบบจัดการคลังสินค้ามีฟังก์ชันพื้นฐาน 15 ฟังก์ชัน ดังนี้

1. เพิ่มสินค้า
2. ดูสินค้า
3. แก้ไขสินค้า
4. ลบสินค้า
5. ขายสินค้า
6. สั่งซื้อสินค้า
7. ดูรายละเอียดการขาย
8. แก้ไขรายละเอียดการขาย
9. ลบรายการขาย
10. ดูรายละเอียดการสั่งซื้อ
11. แก้ไขรายละเอียดการสั่งซื้อ
12. ลบรายการสั่งซื้อ
13. เมนูกลางระบบจัดการคลังสินค้า
14. แสดงผลของ Report ระบบจัดการคลังสินค้า
15. เมนูออกจากโปรแกรม

1.2.2 ระบบจัดการคลังสินค้าประกอบด้วย 4 ไฟล์ ได้แก่

1. Products.bin
2. Sales.bin
3. Purchases.bin
4. Report.bin

1.2.3 ระบบจัดการคลังสินค้ามีการจัดเก็บข้อมูลสินค้าไว้ใน Text File ชื่อ report ซึ่งมี รหัสการขาย ชื่อสินค้า ประเภทสินค้า ราคาขาย จำนวนที่ขาย จำนวนคงเหลือ วันที่ขายสินค้าล่าสุด วันที่สั่งซื้อสินค้าล่าสุด สถานะคลังสินค้า

1.2.4 ระบบจัดการคลังสินค้าจะมีเมนูเพื่อให้ผู้ใช้สามารถเลือก ดำเนินการได้

1.3 ประโยชน์ที่ได้รับ

1.3.1 พัฒนาระบบที่สามารถทำการจัดการคลังสินค้าได้อย่างมีประสิทธิภาพ

1.3.2 พัฒนาทักษะการเขียนโปรแกรม

1.3.3 เรียนรู้การจัดการข้อมูลและไฟล์

1.3.4 เรียนรู้การทำงานร่วมกันเป็นทีม

1.4 เครื่องมือที่คาดว่าจะต้องใช้

1.4.1 โปรแกรม Visual Studio Code

1.4.2 Microsoft Office

บทที่ 2
ระบบจัดการคลังสินค้า

2.1 แฟ้มข้อมูล Product.bin

การจัดการคลังสินค้าของโครงงานนี้ประกอบด้วย 7 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด	ตัวอย่าง
Product ID	VARCHAR	6	P001
Name	VARCHAR	200	น้ำดื่ม ตราคริสตัล
Category	VARCHAR	50	เครื่องดื่ม
Quantity	INTEGER	4	75
Unit	VARCHAR	50	ขวด
Sell Price	FLOAT	4	14.00
Status	VARCHAR	15	Active, Restock

ตารางที่ 2.1 แฟ้มข้อมูลสินค้า

2.1.1 Product ID

Product ID เป็นรหัสที่ใช้ในการระบุสินค้าแต่ละรายการอย่างชัดเจนและไม่ซ้ำกัน ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 6 ตัวอักษร เช่น "P001", "P002" การมีรหัสสินค้าเป็นเอกลักษณ์ช่วยหลีกเลี่ยงความสับสนระหว่างสินค้าหลายรายการ และทำให้การค้นหา แก้ไข หรือติดตามข้อมูลสินค้าเป็นไปได้อย่างแม่นยำและรวดเร็ว

2.1.2 Name

Name คือชื่อเต็มของสินค้าแต่ละรายการ ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 200 ตัวอักษร เช่น "น้ำดื่ม ตราคริสตัล" การมีชื่อสินค้าช่วยให้ผู้ใช้สามารถระบุสินค้าได้อย่างชัดเจน ใช้ตรวจสอบข้อมูลการขาย การสั่งซื้อ และการจัดการสินค้าอื่น ๆ ได้อย่างสะดวก

2.1.3 Category

Category คือประเภทหรือหมวดหมู่ของสินค้า ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 50 ตัวอักษร เช่น "เครื่องดื่ม" การกำหนดประเภทสินค้าช่วยให้สามารถจัดกลุ่มสินค้าได้ง่ายต่อการจัดเก็บ การค้นหา และการออกรายงาน

2.1.4 Quantity

Quantity คือจำนวนสินค้าที่มีอยู่ในคลัง ฟิลด์นี้เป็นประเภทข้อมูลจำนวนเต็ม (INTEGER) ขนาด 4 หลัก เช่น 75 หน่วย การมีจำนวนสินค้าช่วยให้สามารถติดตามสถานะคงคลังได้ชัดเจน และใช้ในการตัดสินใจว่าสินค้าต้องทำการสั่งซื้อเพิ่มเติมหรือไม่

2.1.5 Unit

Unit คือหน่วยที่ใช้ในการนับหรือจำหน่ายสินค้า ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 50 ตัวอักษร เช่น "ขวด" การกำหนดหน่วยนับสินค้าอย่างชัดเจนช่วยให้การบริหารจัดการสินค้าและการบันทึกข้อมูลเป็นไปอย่างถูกต้อง

2.1.6 Sell Price

Sell Price คือราคาที่ใช้ในการจำหน่ายสินค้าแต่ละหน่วย ฟิลด์นี้เป็นประเภทข้อมูลทศนิยม (FLOAT) ขนาด 4 หลัก เช่น 14.00 บาท การมีราคาขายช่วยให้ระบบสามารถคำนวณรายได้จากการขาย และนำข้อมูลไปใช้ในการสร้างรายงานสรุปยอดขายได้

2.1.7 Status

Status คือสถานะของสินค้าในคลัง ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 15 ตัวอักษร เช่น "Active" หรือ "Restock" การมีสถานะสินค้าช่วยให้ผู้ใช้สามารถทราบได้ทันทีว่าสินค้ายังมีจำหน่ายอยู่หรือจำเป็นต้องสั่งซื้อเพิ่ม

2.2 แฟ้มข้อมูล Sales.bin

ไฟล์ Sales.bin ใช้สำหรับจัดเก็บข้อมูลที่เกี่ยวข้องกับการขายสินค้า ประกอบด้วย 6 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดมีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด	ตัวอย่าง
SaleDetail ID	VARCHAR	6	S001
Product ID	VARCHAR	6	อ้างอิงจาก Product.bin.Product ID
Quantity	INTEGER	4	2
Total	FLOAT	4	28.00
Created_at	DATETIME	8	2025-09-12 18:35:42
Updated_at	DATETIME	8	2025-09-12 18:40:42

ตารางที่ 2.2 แฟ้มข้อมูลการขายสินค้า

2.2.1 SaleDetail ID

SaleDetail ID เป็นรหัสที่ใช้ระบุการขายแต่ละครั้งอย่างชัดเจนและไม่ซ้ำกัน ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 6 ตัวอักษร เช่น "S001", "S002" การมีรหัสการขายที่เป็นเอกลักษณ์ช่วยให้สามารถติดตามธุรกรรมการขายได้สะดวกและลดความสับสน

2.2.2 Product ID

Product ID เป็นรหัสที่อ้างอิงมาจากไฟล์ Products.bin โดยใช้ฟิลด์ Product ID ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 6 ตัวอักษร เช่น "P001" การกำหนดรหัสสินค้าในการขายทำให้สามารถเชื่อมโยงข้อมูลการขายกับรายละเอียดของสินค้าได้อย่างถูกต้อง

2.2.3 Quantity

Quantity คือจำนวนสินค้าที่ถูกขายออกไปในแต่ละครั้ง ฟิลด์นี้เป็นประเภทข้อมูลจำนวนเต็ม (INTEGER) ขนาด 4 หลัก เช่น 2 การระบุจำนวนที่ขายช่วยให้ระบบสามารถปรับปรุงจำนวนสินค้าคงเหลือในคลังได้อย่างถูกต้อง

2.2.4 Total

Total คือผลรวมของราคาการขายสินค้าตามจำนวนที่ขาย ฟิลด์นี้เป็นประเภทข้อมูลทศนิยม (FLOAT) ขนาด 4 หลัก เช่น 28.00 บาท การบันทึกราคารวมช่วยให้ผู้ใช้งานสามารถตรวจสอบยอดขายและสรุปผลทางการเงินได้อย่างมีประสิทธิภาพ

2.2.5 Created_at

Created_at คือวันและเวลาที่ทำการบันทึกธุรกรรมการขาย ฟิลด์นี้เป็นประเภทข้อมูลวันและเวลา (DATETIME) ขนาด 8 ไบต์ เช่น 2025-09-12 18:35:42 ข้อมูลนี้มีความสำคัญในการตรวจสอบลำดับเหตุการณ์และประวัติการขาย

2.2.6 Updated_at

Updated_at คือวันและเวลาที่มีการปรับปรุงหรือแก้ไขข้อมูลการขาย ฟิลด์นี้เป็นประเภทข้อมูลวันและเวลา (DATETIME) ขนาด 8 ไบต์ เช่น 2025-09-12 18:40:42 การบันทึกข้อมูลนี้ช่วยให้สามารถติดตามการเปลี่ยนแปลงและตรวจสอบประวัติการแก้ไขได้อย่างโปร่งใส

2.3 แฟ้มข้อมูล Purchases.bin

ไฟล์ Purchases.bin ใช้สำหรับจัดเก็บข้อมูลที่เกี่ยวข้องกับการสั่งซื้อสินค้าเข้าสู่คลังประกอบด้วย 7 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดมีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด	ตัวอย่าง
Purchase ID	VARCHAR	6	PC001
Supplier ID	VARCHAR	6	อ้างอิงจาก product.bin.Product ID
Supplier Name	VARCHAR	50	อ้างอิงจาก product.bin.Name
Total	FLOAT	4	1800.00
Note	VARCHAR	255	ส่งรอบเช้า
Created_at	DATETIME	8	2025-09-12 18:35:42
Updated_at	DATETIME	8	2025-09-12 18:40:42

ตารางที่ 2.3 แฟ้มข้อมูลการสั่งซื้อสินค้าเข้าสู่คลัง

2.3.1 Purchase ID

Purchase ID เป็นรหัสที่ใช้ระบุการสั่งซื้อสินค้าแต่ละครั้งอย่างชัดเจนและไม่ซ้ำกัน ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 6 ตัวอักษร เช่น "PC001", "PC002" การมีรหัสการสั่งซื้อเป็นเอกลักษณ์ช่วยให้สามารถติดตามและอ้างอิงประวัติการสั่งซื้อสินค้าได้สะดวกและลดความสับสน

2.3.2 Product ID

Product ID เป็นรหัสที่อ้างอิงมาจากไฟล์ Products.bin โดยใช้ฟิลด์ Product ID ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 6 ตัวอักษร เช่น "P001" การกำหนดรหัสสินค้าในรายการสั่งซื้อช่วยให้ระบบสามารถเชื่อมโยงข้อมูลกับรายละเอียดสินค้าที่ถูกสั่งซื้อได้อย่างถูกต้อง

2.3.3 Product ID

Quantity คือจำนวนสินค้าที่ถูกสั่งซื้อสู่คลังในแต่ละครั้ง ฟิลด์นี้เป็นประเภทข้อมูลจำนวนเต็ม (INTEGER) ขนาด 4 หลัก เช่น 180 การระบุจำนวนที่สั่งซื้อมีความสำคัญต่อการปรับปรุงสต็อกสินค้าในคลังให้ถูกต้องและเป็นปัจจุบัน

2.3.4 Total

Total คือราคารวมทั้งหมดของสินค้าที่สั่งซื้อในแต่ละครั้ง ฟิลด์นี้เป็นประเภทข้อมูลทศนิยม (FLOAT) ขนาด 4 หลัก เช่น 1800.00 บาท การบันทึกราคารวมช่วยให้ผู้ใช้งานสามารถตรวจสอบต้นทุนการสั่งซื้อสินค้าและนำไปวิเคราะห์ต้นทุน-กำไรได้

2.3.5 Note

Note คือข้อความเพิ่มเติมที่ใช้บันทึกรายละเอียดเกี่ยวกับการสั่งซื้อสินค้า ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (VARCHAR) ขนาด 255 ตัวอักษร เช่น "ส่งรอบเช้า" การมีช่องหมายเหตุช่วยให้ผู้ใช้งานสามารถเก็บข้อมูลรายละเอียดเสริมของการสั่งซื้อเพื่อใช้ตรวจสอบภายหลังได้

2.3.6 Created_at

Created_at คือวันและเวลาที่ทำการบันทึกการสั่งซื้อสินค้า ฟิลด์นี้เป็นประเภทข้อมูลวันและเวลา (DATETIME) ขนาด 8 ไบต์ เช่น 2025-09-12 18:35:42 ข้อมูลนี้ช่วยให้สามารถตรวจสอบและติดตามประวัติการสั่งซื้อสินค้าได้อย่างเป็นลำดับเวลา

2.3.7 Updated_at

Updated_at คือวันและเวลาที่มีการปรับปรุงหรือแก้ไขข้อมูลการสั่งซื้อ ฟิลด์นี้เป็นประเภทข้อมูลวันและเวลา (DATETIME) ขนาด 8 ไบต์ เช่น 2025-09-12 18:40:42 การบันทึกข้อมูลนี้ช่วยเพิ่มความโปร่งใสและตรวจสอบการแก้ไขย้อนหลังได้

2.4 แฟ้มข้อมูล Report.bin

ไฟล์ Report.bin ใช้สำหรับจัดเก็บข้อมูลสรุปที่ดึงมาจากหลายตาราง (Products, Sales และ Purchases) เพื่อนำมาแสดงในรูปแบบรายงาน โดยประกอบด้วย 9 ฟิลด์หลัก ดังนี้

```
                                   ==================================
                                            STOCK MANAGEMENT REPORT
                                   ==================================

Generated At : 2025-10-02 00:48:10 (+07:00)
App Version : 1.0
Endianness : Little-Endian
Encoding : UTF-8 (fixed-length)

[1] PRODUCT SUMMARY
+-------+-----------------------------+-------------+---------+------------+-------------+---------------+---------------+-----------+
| SellId | Name                        | Category    | Price   | Sell Amount| Prod Remain | Last Sell     | Last Purchase | Status    |
+-------+-----------------------------+-------------+---------+------------+-------------+---------------+---------------+-----------+
| S00001 | น้ำดื่มสิงห์ 600 มล.        | เครื่องดื่ม  |    7.00 |         20 |          60 |   24/09 20:50 |         24/09 | Active    |
| S00002 | ขนมปังเลอแปงไส้ครีม         | ขนม         |   14.00 |         10 |          45 |   01/10 13:24 |         01/10 | Restock   |
| S00003 | น้ำดื่มสิงห์ 600 มล.        | เครื่องดื่ม  |    7.00 |         25 |          60 |   24/09 20:50 |         24/09 | Active    |
| S00004 | ขนมปังเลอแปงไส้ครีม         | ขนม         |   14.00 |          7 |          45 |   01/10 13:24 |         01/10 | Restock   |
+-------+-----------------------------+-------------+---------+------------+-------------+---------------+---------------+-----------+


Summary (Active only)
- Total Products (records) : 3
- Active Products : 2
- Restock Products: 1
- Deactivate Products : 0

Price Statistics (THB, Active only)
- Min Price : 7.00
- Max Price : 63.00

Products by Category (Active only)
- เครื่องดื่ม : 1
- อาหารแห้ง : 1
```

รูปภาพที่ 2-1 ไฟล์ report

2.4.1 Sell ID

Sell ID เป็นรหัสที่อ้างอิงมาจากไฟล์ Sales.bin โดยใช้ฟิลด์ SaleDetail ID ฟิลด์นี้ช่วยให้สามารถเชื่อมโยงข้อมูลรายงานกับธุรกรรมการขายแต่ละครั้งได้อย่างถูกต้อง

2.4.2 Name

Name เป็นชื่อสินค้าที่อ้างอิงมาจากไฟล์ Products.bin โดยใช้ฟิลด์ Name ข้อมูลนี้ช่วยให้สามารถระบุสินค้าในรายงานได้อย่างชัดเจนและเข้าใจง่าย

2.4.3 Category

Category คือประเภทของสินค้าที่อ้างอิงมาจากไฟล์ Products.bin โดยใช้ฟิลด์ Category การมีประเภทสินค้าในรายงานช่วยให้สามารถจัดกลุ่มและวิเคราะห์ข้อมูลสินค้าได้สะดวก

2.4.4 Price

Price คือราคาขายของสินค้า โดยอ้างอิงจากไฟล์ Products.bin โดยใช้ฟิลด์ Sell Price ข้อมูลนี้มีความสำคัญต่อการคำนวณรายได้และการวิเคราะห์กำไรของระบบ

2.4.5 Sell Amount

Sell Amount คือจำนวนสินค้าที่ถูกขายออก โดยอ้างอิงจากไฟล์ Sales.bin โดยใช้ฟิลด์ Quantity ซึ่งช่วยให้ผู้ใช้งานสามารถติดตามยอดขายของสินค้าแต่ละรายการได้

2.4.6 Prod Remain

Prod Remain คือจำนวนสินค้าที่ยังคงเหลืออยู่ในคลัง โดยอ้างอิงจากไฟล์ Products.bin โดยใช้ฟิลด์ Quantity ข้อมูลนี้ช่วยให้ผู้ใช้งานสามารถประเมินสต็อกสินค้าได้อย่างถูกต้อง

2.4.7 Last Sell

Last Sell คือวันและเวลาที่มีการขายสินค้าล่าสุด โดยอ้างอิงจากไฟล์ Sales.bin โดยใช้ฟิลด์ Created_at ข้อมูลนี้ช่วยในการตรวจสอบพฤติกรรมการขายและความเคลื่อนไหวของสินค้า

2.4.8 Last Purchase

Last Purchase คือวันและเวลาที่มีการสั่งซื้อสินค้าล่าสุด โดยอ้างอิงจากไฟล์ Purchases.bin โดยใช้ฟิลด์ Created_at ข้อมูลนี้ช่วยให้ผู้ใช้งานสามารถตรวจสอบความถี่ในการสั่งซื้อสินค้าได้

2.4.9 Status

Status คือสถานะของสินค้า โดยอ้างอิงจากไฟล์ Products.bin โดยใช้ฟิลด์ Status เช่น "Active" หรือ "Restock" เพื่อแสดงว่าสินค้าพร้อมจำหน่ายหรือจำเป็นต้องสั่งซื้อเพิ่ม

บทที่ 3
การใช้งานระบบจัดการคลังสินค้า

โปรแกรมระบบจัดการคลังสินค้าเป็นระบบที่พัฒนาขึ้นเพื่อช่วยให้การจัดเก็บและบริหารข้อมูลสินค้าเป็นไปอย่างสะดวก รวดเร็ว และมีประสิทธิภาพ โดยผู้ใช้สามารถบันทึกข้อมูลสินค้า การขายสินค้า การสั่งซื้อสินค้า รวมถึงสามารถแสดงรายงานสรุปที่เกี่ยวข้องได้อย่างเป็นระบบ

โปรแกรมระบบจัดการคลังสินค้าประกอบไปด้วยการเพิ่มข้อมูลสินค้า ซึ่งจะเก็บรายละเอียดเช่น รหัสสินค้า (Product ID), ชื่อสินค้า (Name), ประเภทสินค้า (Category), จำนวนคงเหลือ (Quantity), หน่วยสินค้า (Unit), ราคาขาย (Sell Price) และสถานะสินค้า (Status) ผู้ใช้สามารถแสดงข้อมูลสินค้าทั้งหมดในโปรแกรม

ในส่วนของการอัพเดตข้อมูล ผู้ใช้สามารถแก้ไขหรือเปลี่ยนแปลงรายละเอียดสินค้าได้ หากไม่ต้องการแก้ไขบางส่วนก็สามารถกด Enter เพื่อข้ามไปยังข้อมูลถัดไปได้ทันที สำหรับการลบข้อมูลสามารถลบโดยใช้รหัสสินค้า (Product ID) เพื่อทำการลบข้อมูลทั้งหมดที่เกี่ยวข้องกับสินค้านั้น

นอกจากนี้ ระบบยังสามารถสร้างรายงานสรุปการสั่งซื้อ–ขายสินค้า รวมถึงจำนวนสินค้าคงเหลือ เพื่อช่วยในการตรวจสอบและวิเคราะห์ข้อมูลเชิงธุรกิจได้อย่างครบถ้วน เมื่อสิ้นสุดการใช้งาน ผู้ใช้สามารถบันทึกข้อมูลและออกจากโปรแกรมได้อย่างปลอดภัย

สำหรับผู้ใช้งานโปรแกรม

3.1 การใช้งานโปรแกรมระบบจัดการคลังสินค้า

3.1.1 กรอกหมายเลข 1 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน products.bin เพิ่มข้อมูลที่ประกอบไปด้วย Add product, Read product, Read all products, Read by Id, Update product, Delete product, Delete specific product, Delete all products, Sell product, Purchase product, Exit

```
Enter file name

1: products.bin
2: sales.bin
3: purchases.bin
4: print_report.txt
0: Exit

your input: 1
```

รูปภาพที่ 3-1 การเลือกใช้งานฟังก์ชั่น สินค้า

3.1.2 เมื่อเมนูฟังก์ชัน products.bin ขึ้นมาแล้วจากนั้นก็สามารถระบุเมนูที่ต้องการเลือกได้

```
Please choose from list:

| 1. Add product
| 2. Read product
|-- 2.1 Read all products
|-- 2.2 Read by Id
| 3. Update product
| 4. Delete product
|-- 4.1 Delete specific product
|-- 4.2 Delete all products
| 5. Sell product
| 6. Purchase product
| 0. Exit

your input: 2
```

รูปภาพที่ 3-2 เมนูของ สินค้า

3.1.3 กรอกหมายเลข 2 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน products.bin เพิ่มข้อมูลที่ประกอบไปด้วย Add product, Read product, Read all products, Read by Id, Update product, Delete product, Delete specific product, Delete all products, Sell product, Purchase product, Exit

```
Enter file name

1: products.bin
2: sales.bin
3: purchases.bin
4: print_report.txt
0: Exit

your input: 2
```

รูปภาพที่ 3-3 การเลือกใช้งานฟังก์ชั่น การขายสินค้า

3.1.4 เมื่อเมนูฟังก์ชั่น sales.bin ขึ้นมาแล้วจากนั้นก็สามารถระบุเมนูที่ต้องการ เลือกได้

```
Please choose from list:

| 1. Read sale details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update sale details by Id
| 3. Delete sale details
|-- 3.1 Delete specific sale
|-- 3.2 Delete all sales
| 0. Exit

your input: 1
```

รูปภาพที่ 3-4 เมนูของ การขายสินค้า

3.1.5 กรอกหมายเลข 3 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน purchases.bin เพิ่มข้อมูลที่ประกอบไปด้วย Read purchase details, Read all fields, Read by Id, Update purchase, details by Id, Delete purchase details, Delete specific purchase, Delete all purchases, Exit

```
Enter file name

1: products.bin
2: sales.bin
3: purchases.bin
4: print_report.txt
0: Exit

your input: 3
```

รูปภาพที่ 3-5 การเลือกใช้งานฟังก์ชั่น คลังสินค้า

3.1.6 เมื่อเมนูฟังก์ชั่น purchases.bin ขึ้นมาแล้วจากนั้นก็สามารถระบุเมนูที่ต้องการเลือกได้

```
Please choose from list:

| 1. Read purchase details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update purchase details by Id
| 3. Delete purchase details
|-- 3.1 Delete specific purchase
|-- 3.2 Delete all purchases
| 0. Exit

your input: 1
```

รูปภาพที่ 3-6 เมนูของ คลังสินค้า

3.1.7 กรอกหมายเลข 4 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Generate report เพื่อเพิ่ม ไฟล์ report.txt ที่สามารถเขียน report ระบบจัดการคลังสินค้าได้

```
your input: 0

Enter file name

1: products.bin
2: sales.bin
3: purchases.bin
4: print_report.txt
0: Exit

your input: 4
```

รูปภาพที่ 3-7 การเลือกใช้งานฟังก์ชั่น การรายงาน

3.1.8 กรอกหมายเลข 0 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Exit เพื่อออกจากโปรแกรม

```
Enter file name

1: products.bin
2: sales.bin
3: purchases.bin
4: print_report.txt
0: Exit

your input: 5
```

รูปภาพที่ 3-8 การเลือกใช้งานฟังก์ชั่นของ Exit

3.2 การใช้งานโปรแกรมเพิ่มข้อมูล

3.2.1 กรอกหมายเลข 1 เพื่อเพิ่มข้อมูลทั้งหมดของสินค้าที่มีในโปรแกรม

```
Enter file name

1: products.bin
2: sales.bin
3: purchases.bin
4: print_report.txt
0: Exit

your input: 1
```

รูปภาพที่ 3-9 การเลือกใช้งานฟังก์ชั่น Add product

3.2.2 เมื่อกดเลือกหมายเลข 1 จะปรากฏหัวข้อการใส่รหัสหนังสือและจากนั้นใส่ข้อมูลในหัวข้อทั้งหมดดังภาพที่ 3-10

```
Product added successfully:
ID: P0093
Name: Mama tumyam
Category: อาหารแห้ง
Quantity: 5
Unit: ห่อ
Price: 7.00
Status: Deactive

Do you want to continue? (0 for no, other for yes): 1
```

รูปภาพที่ 3- 10 การเพิ่มสินค้า

3.3 การใช้งานโปรแกรมแสดงข้อมูล

3.3.1 กรอกหมายเลข 2 เพื่อแสดงข้อมูลทั้งหมดของสินค้าที่มีในโปรแกรม

```
Please choose from list:

| 1. Add product
| 2. Read product
|-- 2.1 Read all products
|-- 2.2 Read by Id
| 3. Update product
| 4. Delete product
|-- 4.1 Delete specific product
|-- 4.2 Delete all products
| 5. Sell product
| 6. Purchase product
| 0. Exit

your input: 2
```

รูปภาพที่ 3- 11 การเลือกใช้งานฟังก์ชั่น Read product

3.3.2 เมื่อกดเลือกหมายเลข 2 จะปรากฏข้อมูลตัวเลือกในการแสดงสินค้าทั้งหมดและแสดงเฉพาะสินค้าที่เลือกโดยรหัสสินค้าดังภาพที่ 3-12

```
Please choose from list:

| 1. Add product
| 2. Read product
|-- 2.1 Read all products
|-- 2.2 Read by Id
| 3. Update product
| 4. Delete product
|-- 4.1 Delete specific product
|-- 4.2 Delete all products
| 5. Sell product
| 6. Purchase product
| 0. Exit

your input: 2

Read Options:
1. Read all products
2. Read specific product by ID

Enter option (1-2): 1
```

รูปภาพที่ 3-12 แสดงข้อมูล Read product

3.3.3 กรอกหมายเลข 1 จะปรากฏข้อมูลสินค้าทั้งหมดดังภาพที่ 3-13

```
Read Options:
1. Read all products
2. Read specific product by ID

Enter option (1-2): 1

Product List:
==========================================================================================
| Product ID | Name                          | Category    | Quantity    | Unit    | Price       | Status      |
==========================================================================================
| P0092         Mama                                 อาหารแห้ง   5           ห่อ         7.00        Deactive
| P0093         Mama tumyam                          อาหารแห้ง   5           ห่อ         7.00        Deactive
==========================================================================================

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3-13 แสดงข้อมูล Read all product

3.3.4 กรอกหมายเลข 2 จะปรากฏข้อมูลสินค้าที่เลือก โดยรหัสสินค้าดังภาพที่ 3-14

```
Read Options:
1. Read all products
2. Read specific product by ID

Enter option (1-2): 2
Enter product ID (start with P): P0092

Product P0092 found:
=======================================================
| Product ID : P0092
| Name       : Mama
| Category   : อาหารแห้ง
| Quantity   : 5
| Unit       : ห่อ
| Price      : 7.00
| Status     : Deactive
=======================================================

Do you want to continue? (0 for no, other for yes): 1
```

รูปภาพที่ 3-14 แสดงข้อมูล สินค้าที่เลือก โดยรหัสสินค้า

3.3.5 กรอกหมายเลข 1 เพื่อแสดงข้อมูลทั้งหมดของการขายสินค้า

```
Please choose from list:

| 1. Read sale details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update sale details by Id
| 3. Delete sale details
|-- 3.1 Delete specific sale
|-- 3.2 Delete all sales
| 0. Exit

your input:
```

รูปภาพที่ 3-15 การเลือกใช้งานฟังก์ชั่น Read sale details

3.3.6 เมื่อกดเลือกหมายเลข 1 จะปรากฏข้อมูลตัวเลือกในการแสดงการขายสินค้าทั้งหมดและการแสดงการขายสินค้าเฉพาะรหัสสินค้าที่ต้องการดังภาพที่ 3-16

```
Please choose from list:

| 1. Read sale details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update sale details by Id
| 3. Delete sale details
|-- 3.1 Delete specific sale
|-- 3.2 Delete all sales
| 0. Exit

your input: 1

1: Read all fields
2: Read by Id
Your input: 1
```

รูปภาพที่ 3- 16 แสดงเมนู Read sale details

3.3.7 เมื่อกดเลือกหมายเลข 1 จะปรากฏข้อมูลการขายสินค้าทั้งหมดดังภาพที่ 3-17

```
1: Read all fields
2: Read by Id
Your input: 1
==================================================================================================
| Sale ID  | Product ID  | Quantity  | Total Amount        | Created At     | Updated At     |
==================================================================================================
| S00001   | P00001      | 20        | 145.0               | 1758721795.46  | 1758721795.46  |
| S00002   | P00002      | 10        | 145.0               | 1758721808.44  | 1758721808.44  |
| S00003   | P00001      | 25        | 180.0               | 1758721822.15  | 1758721822.15  |
| S00004   | P00002      | 7         | 98.0                | 1759299895.66  | 1759299895.66  |
==================================================================================================

Do you want to continue? (0 for no, other for yes): 1
```

รูปภาพที่ 3- 17 แสดงข้อมูล Read all fields

3.3.8 เมื่อกดเลือกหมายเลข 2 จะปรากฏข้อมูลการขายสินค้าเฉพาะรหัสที่ต้องการดังภาพที่ 3-18

```
1: Read all fields
2: Read by Id
Your input: 2

Enter Sale ID: S00001
Sale details for ID S00001:
=======================================================
| Sale ID      : S00001
| Product ID   : P00001
| Quantity     : 20
| Total Amount : 145.0
| Created At   : 1758721795.46
| Updated At   : 1758721795.46
=======================================================

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 18 แสดงข้อมูล Read Id

3.3.9 กรอกหมายเลข 1 เพื่อแสดงข้อมูลทั้งหมดของคลังสินค้า

```
Please choose from list:

| 1. Read purchase details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update purchase details by Id
| 3. Delete purchase details
|-- 3.1 Delete specific purchase
|-- 3.2 Delete all purchases
| 0. Exit

your input:
```

รูปภาพที่ 3- 19 แสดงข้อมูล Read purchase details

3.3.10 เมื่อกดเลือกหมายเลข 1 จะปรากฏตัวเลือกการแสดงข้อมูลของคลังสินค้าทั้งหมดและการแสดงข้อมูลของคลังสินค้าโดยรหัสดังภาพที่ 3-20

```
Please choose from list:

| 1. Read purchase details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update purchase details by Id
| 3. Delete purchase details
|-- 3.1 Delete specific purchase
|-- 3.2 Delete all purchases
| 0. Exit

your input: 1

1: Read all fields
2: Read by Id
Your input: 1
```

รูปภาพที่ 3- 20 แสดงเมนู Read purchase details

3.3.11 เมื่อกดเลือกหมายเลข 1 จะปรากฏข้อมูลของคลังสินค้าทั้งหมดดังภาพที่ 3-21

```
1: Read all fields
2: Read by Id
Your input: 1
=================================================================================================================
| Purchase ID    | Product ID    | Quantity   | Total      | Note               | Created At      | Updated At      |
=================================================================================================================
| I00001         | P00003        | 11         | 693.0      | ส่งรอบเช้า          | 1758721836.00   | 1758721836.00   |
| I00002         | P00001        | 5          | 35.0       | ส่งรอบเช้า          | 1758721853.52   | 1758721853.52   |
| I00003         | P00002        | 10         | 140.0      | ส่งรอบเช้า          | 1758721862.29   | 1758721862.29   |
| I00004         | P00002        | 5          | 60.0       | Morning routine    | 1759299919.80   | 1759299919.80   |
=================================================================================================================

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 21 แสดงข้อมูล Read all fields

3.3.12 เมื่อกดเลือกหมายเลข 2 จะปรากฏข้อมูลของคลังสินค้าโดยรหัสดังภาพที่ 3-22

```
1: Read all fields
2: Read by Id
Your input: 2

Enter Purchase ID: I00001
Purchase details for ID I00001:
=======================================================
| Purchase ID : I00001
| Product ID  : P00003
| Quantity    : 11
| Total       : 693.0
| Note        : ส่งรอบเช้า
| Created At  : 1758721836.00
| Updated At  : 1758721836.00
=======================================================

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 22 แสดงข้อมูล Read by Id

3.4 การใช้งานโปรแกรมแก้ไขข้อมูล

3.4.1 กรอกหมายเลข 3 เพื่อแก้ไขข้อมูลของสินค้าที่มีในโปรแกรม

```
Please choose from list:

| 1. Add product
| 2. Read product
|-- 2.1 Read all products
|-- 2.2 Read by Id
| 3. Update product
| 4. Delete product
|-- 4.1 Delete specific product
|-- 4.2 Delete all products
| 5. Sell product
| 6. Purchase product
| 0. Exit

your input: 3
```

รูปภาพที่ 3- 23 การเลือกใช้งานฟังก์ชั่น Update product

3.4.2 เมื่อกดเลือกหมายเลข 3 จะปรากฏให้แก้ไขข้อมูลสินค้าโดยรหัสดังภาพที่ 3-23

```
your input: 3
Enter product ID (start with P): P0093

Current Product Values:
+-------------------------------------------------+
| 1. Name     : Mama tumyam
| 2. Category : อาหารแห้ง                          |
| 3. Quantity : 5
| 4. Unit     : ห่อ                                |
| 5. Price    : 7.00
+-------------------------------------------------+

Update Options:
1. Name
2. Category
3. Quantity
4. Unit
5. Sell Price
6. Update all fields
0. Cancel update

Enter field number to update (0-6): 3
Enter new quantity: 50

Product P0093 updated successfully:
Name: Mama tumyam
Category: อาหารแห้ง
Quantity: 50
Unit: ห่อ
Price: 7.00
Status: Active

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 24 การแก้ไขข้อมูล Update product

3.4.3 กรอกหมายเลข 2 เพื่อแก้ไขข้อมูลการขายสินค้าโดยรหัสที่มีในโปรแกรม

```
Please choose from list:

| 1. Read sale details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update sale details by Id
| 3. Delete sale details
|-- 3.1 Delete specific sale
|-- 3.2 Delete all sales
| 0. Exit

your input: 2
```

รูปภาพที่ 3- 25 การเลือกใช้งานฟังก์ชั่น Update sale details by Id

3.4.4 เมื่อกดเลือกหมายเลข 2 จะปรากฏให้แก้ไขข้อมูลการขายสินค้าโดยรหัสดังภาพที่ 3-25

```
Please choose from list:

| 1. Read sale details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update sale details by Id
| 3. Delete sale details
|-- 3.1 Delete specific sale
|-- 3.2 Delete all sales
| 0. Exit

your input: 2
Enter Sale ID to update: S00001
Current values:
| Sale ID      : S00001
| Product ID   : P00001
| Quantity     : 20
| Total Amount : 145.0
| Created At   : 1758721795.46
| Updated At   : 1758721795.46
Enter new quantity (or -1 to keep): 1
Enter new total amount (or -1 to keep): -1
Sale record updated.

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 26 การแก้ไขข้อมูล Update sale details by Id

3.4.5 กรอกหมายเลข 2 เพื่อแก้ไขข้อมูลคลังสินค้าโดยรหัสที่มีในโปรแกรม

```
Please choose from list:

| 1. Read purchase details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update purchase details by Id
| 3. Delete purchase details
|-- 3.1 Delete specific purchase
|-- 3.2 Delete all purchases
| 0. Exit

your input: 2
```

รูปภาพที่ 3- 27 การเลือกใช้งานฟังก์ชั่น Update purchase details by Id

3.4.6 เมื่อกดเลือกหมายเลข 2 จะปรากฏให้แก้ไขข้อมูลคลังสินค้าโดยรหัสดังภาพที่ 3-27

3.5 การใช้งานโปรแกรมลบข้อมูล

3.5.1 กรอกหมายเลข 4 เพื่อลบสินค้าที่มีในโปรแกรม

```
Please choose from list:

| 1. Read purchase details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update purchase details by Id
| 3. Delete purchase details
|-- 3.1 Delete specific purchase
|-- 3.2 Delete all purchases
| 0. Exit

your input: 2
Enter Purchase ID to update: I00001
Current values:
| Purchase ID : I00001
| Product ID  : P00003
| Quantity    : 11
| Total       : 693.0
| Note        : ส่งรอบเช้า
| Created At  : 1758721836.00
| Updated At  : 1758721836.00
Enter new quantity (or -1 to keep): 15
Enter new total (or -1 to keep): 700.0
Enter new note (or leave blank to keep):
Purchase record updated.

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 28 การแก้ไขข้อมูล Update purchare details by Id

```
Please choose from list:

| 1. Add product
| 2. Read product
|-- 2.1 Read all products
|-- 2.2 Read by Id
| 3. Update product
| 4. Delete product
|-- 4.1 Delete specific product
|-- 4.2 Delete all products
| 5. Sell product
| 6. Purchase product
| 0. Exit

your input: 4
```

รูปภาพที่ 3- 29 การเลือกใช้งานฟังก์ชั่น Delete product

3.5.2 เมื่อกดเลือกหมายเลข 4 จะปรากฏตัวเลือกการลบข้อมูลสินค้าโดยรหัสและการลบสินค้าทั้งหมดดังภาพที่ 3-29

```
Please choose from list:

| 1. Add product
| 2. Read product
|-- 2.1 Read all products
|-- 2.2 Read by Id
| 3. Update product
| 4. Delete product
|-- 4.1 Delete specific product
|-- 4.2 Delete all products
| 5. Sell product
| 6. Purchase product
| 0. Exit

your input: 4

Delete Options:
1. Delete specific product
2. Delete all products
0. Cancel

Enter option (0-2): 1
```

รูปภาพที่ 3- 30 แสดงเมนู Delete product

3.5.3 เมื่อกดเลือกหมายเลข 1 จะทำการลบข้อมูลสินค้าโดยรหัสดังภาพที่ 3-30

```
Delete Options:
1. Delete specific product
2. Delete all products
0. Cancel

Enter option (0-2): 1
Enter product ID (start with P): P0092

Product to delete:
+-------------------------------------------------+
| Product ID : P0092                              |
| Name       : Mama                               |
| Category   : อาหารแห้ง                          |
| Quantity   : 5                                  |
| Unit       : ห่อ                                |
| Price      : 7.00                               |
| Status     : Deactive                           |
+-------------------------------------------------+

Are you sure you want to delete product P0092? (yes/no): yes

Product P0092 has been deleted successfully.

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 31 การลบข้อมูล Delete specific product

3.5.4 เมื่อกดเลือกหมายเลข 2 จะทำการลบข้อมูลสินค้าทั้งหมดดังภาพที่ 3-31

```
Delete Options:
1. Delete specific product
2. Delete all products
0. Cancel

Enter option (0-2): 2

Are you sure you want to delete ALL products? This action cannot be undone! (yes/no): yes

All products have been deleted successfully.

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 32 การลบข้อมูล Delete all products

3.5.5 กรอกหมายเลข 3 เพื่อลบการขายที่มีในโปรแกรม

```
Please choose from list:

| 1. Read sale details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update sale details by Id
| 3. Delete sale details
|-- 3.1 Delete specific sale
|-- 3.2 Delete all sales
| 0. Exit

your input: 3
```

รูปภาพที่ 3- 33 การเลือกใช้งานฟังก์ชั่น Delete sale details

3.5.6 เมื่อกดเลือกหมายเลข 3 จะปรากฏตัวเลือกการลบการขายสินค้าโดยรหัสและการลบการขายสินค้าทั้งหมดดังภาพที่ 3-33

```
Please choose from list:

| 1. Read sale details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update sale details by Id
| 3. Delete sale details
|-- 3.1 Delete specific sale
|-- 3.2 Delete all sales
| 0. Exit

your input: 3
Delete Options:
1. Delete specific sale
2. Delete all sales
0. Cancel
Your input: 1
```

รูปภาพที่ 3- 34 แสดงเมนู Delete sale details

3.5.7 เมื่อกดเลือกหมายเลข 1 จะทำการลบการขายสินค้าโดยรหัสดังภาพที่ 3-34

```
Delete Options:
1. Delete specific sale
2. Delete all sales
0. Cancel
Your input: 1
Enter Sale ID to delete: S00001
Sale record deleted.

Do you want to continue? (0 for no, other for yes): 1
```

รูปภาพที่ 3- 35 แสดงเมนู Delete all sales

3.5.8 เมื่อกดเลือกหมายเลข 2 จะทำการลบการขายสินค้าทั้งหมดดังภาพที่ 3-35

```
Delete Options:
1. Delete specific sale
2. Delete all sales
0. Cancel
Your input: 2
Are you sure you want to delete all sales? (y/n): y
All sale records deleted.

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 36 แสดงเมนู Delete all details

3.5.9 กรอกหมายเลข 3 เพื่อลบข้อมูลในคลังสินค้าโดยรหัสที่มีในโปรแกรม

```
Please choose from list:

| 1. Read purchase details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update purchase details by Id
| 3. Delete purchase details
|-- 3.1 Delete specific purchase
|-- 3.2 Delete all purchases
| 0. Exit

your input: 3
```

รูปภาพที่ 3- 37 การเลือกใช้งานฟังก์ชั่น Delete purchase details

3.5.10 เมื่อกดเลือกหมายเลข 3 จะปรากฏตัวเลือกการลบข้อมูลในคลังสินค้าโดยรหัสและการลบข้อมูลในคลังสินค้าทั้งหมดดังภาพที่ 3-37

```
Please choose from list:

| 1. Read purchase details
|-- 1.1 Read all fields
|-- 1.2 Read by Id
| 2. Update purchase details by Id
| 3. Delete purchase details
|-- 3.1 Delete specific purchase
|-- 3.2 Delete all purchases
| 0. Exit

your input: 3
Delete Options:
1. Delete specific purchase
2. Delete all purchases
0. Cancel
Your input:
```

รูปภาพที่ 3- 38 แสดงเมนู Delete purchase details

3.5.11 เมื่อกดเลือกหมายเลข 1 จะทำการลบข้อมูลในคลังสินค้าโดยรหัสดังภาพที่ 3-38

```
Delete Options:
1. Delete specific purchase
2. Delete all purchases
0. Cancel
Your input: 1
Enter Purchase ID to delete: I00001
Purchase record deleted.

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 39 การลบ Delete purchase details

3.5.12 เมื่อกดเลือกหมายเลข 2 จะทำการลบข้อมูลในคลังสินค้าทั้งหมดดังภาพที่ 3-39

```
Delete Options:
1. Delete specific purchase
2. Delete all purchases
0. Cancel
Your input: 2
Are you sure you want to delete all purchases? (y/n): y
All purchase records deleted.

Do you want to continue? (0 for no, other for yes):
```

รูปภาพที่ 3- 40 การลบ Delete all purchases

บทที่ 4
อธิบายการทำงานของ Code

4.1 ฟังก์ชั่นไบนารีพื้นฐานในระบบจัดการคลังสินค้า

4.1.1 Module Struct เป็นโมดูลในภาษา Python ที่ใช้สำหรับการจัดการข้อมูลในรูปแบบ ไบนารี เช่น การแปลงข้อมูลจากชนิดข้อมูล Python (เช่น int, float, string) ไปเป็น byte stream และการแปลงกลับจาก byte stream ไปเป็นชนิดข้อมูลเดิม โมดูลนี้มีความสำคัญเมื่อเราต้องการจัดเก็บหรืออ่านข้อมูลจากไฟล์ที่เก็บในรูปแบบ Binary file (.bin)

```
import struct
```

รูปภาพที่ 4- 1 Code Module pickle

4.1.2 import datetime เป็นโมดูลที่ใช้ในการจัดการเกี่ยวกับ วันและเวลา โดยมี class หลัก ๆ เช่น

4.1.2.1 datetime : ใช้แทนวันและเวลารวมกัน (ปี, เดือน, วัน, ชั่วโมง, นาที, วินาที)

4.1.2.2 date : ใช้แทนแค่วันที่ (ปี, เดือน, วัน)time : ใช้แทนแค่เวลา (ชั่วโมง, นาที, วินาที)

4.1.2.3 timedelta : ใช้แทนช่วงเวลาที่สามารถนำไปบวก/ลบกับวันเวลาได้

ในโปรแกรมระบบคลังสินค้า datetime ถูกนำมาใช้เพื่อเก็บข้อมูลวันที่และเวลาที่ทำการบันทึกรายการ เช่น Created_at และ Updated_at ของการขายสินค้า (Sales) และการสั่งซื้อสินค้า (Purchases)

```
import datetime
```

รูปภาพที่ 4- 2 Code Module datetime

4.1.3 Module Struct เป็นโมดูลที่ใช้สำหรับทำงานร่วมกับ ระบบปฏิบัติการ (Operating System) โดยให้ฟังก์ชันที่ช่วยจัดการไฟล์และโฟลเดอร์ เช่น การตรวจสอบ path ปัจจุบัน (os.getcwd()), การสร้าง path แบบ absolute (os.path.abspath()), การตรวจสอบว่าไฟล์มีอยู่จริงหรือไม่ รวมถึงคำสั่งในการสร้าง ลบ หรือเปลี่ยน directory ได้อย่างสะดวก

```
import os
```

รูปภาพที่ 4- 3 Code Module os

4.2 โครงสร้างข้อมูลหลักของโปรแกรม

4.2.1 โครงสร้างข้อมูลสินค้า product_fmt = "<6s200s50si50sf15s" คือ format string ที่ใช้กำหนด layout ของข้อมูลสินค้าแต่ละ record ดังรูปที่ 4-4

```
product_fmt = "<6s200s50si50sf15s"
```

รูปภาพที่ 4- 4 โครงสร้างข้อมูลสินค้า

4.2.2 โครงสร้างข้อมูลการขาย sale_fmt = "<6s6sifdd" คือ format string ที่ใช้กำหนด layout ของข้อมูลข้อมูลการขายแต่ละ record ดังรูปที่ 4-5

```
sale_fmt = "<6s6sifdd"
```

รูปภาพที่ 4- 5 โครงสร้างข้อมูลการขาย

4.2.3 โครงสร้างข้อมูลการสั่งซื้อ purchase_fmt = "<6s6sif255sdd" คือ format string ที่ใช้กำหนด layout ของข้อมูลข้อมูลการสั่งซื้อแต่ละ record ดังรูปที่ 4-6

```
purchase_fmt = "<6s6sif255sdd"
```

รูปภาพที่ 4- 6 โครงสร้างข้อมูลการสั่งซื้อ

4.3 ฟังก์ชันการทำงานเมนูหลัก

4.3.1 ฟังก์ชัน product_handler

ฟังก์ชัน product_handler ทำหน้าที่จัดการข้อมูลสินค้า โดยรองรับการเพิ่มสินค้าใหม่, แก้ไขข้อมูลสินค้าเดิม, ค้นหาสินค้า, ลบสินค้า และแสดงรายการสินค้าทั้งหมด ข้อมูลสินค้าจะถูกจัดเก็บในไฟล์ products.bin ในรูปแบบ Binary Record ตามโครงสร้างที่กำหนดใน product_fmt โครงสร้างข้อมูล (product_fmt = "<6s200s50si50sf15s") ประกอบด้วย

1) รหัสสินค้า (Product ID)
2) ชื่อสินค้า (Name)
3) ประเภทสินค้า (Category)
4) จำนวนสินค้า (Quantity)
5) หน่วยนับ (Unit)
6) ราคาขาย (Sell Price)
7) สถานะ (Status)

ข้อมูลแต่ละเรคอร์ดจะถูก pack/unpack ด้วยโมดูล struct เพื่อให้ทุกเรคอร์ดมีขนาดเท่ากัน ทำให้สามารถอ่าน–เขียนข้อมูลได้สะดวกและมีประสิทธิภาพ

```
def product_handler(path, fmt, size):
    while True:
        try:
            print("\nPlease choose from list:")
            print("\n| 1. Add product")
            print("| 2. Read product")
            print("|-- 2.1 Read all products")
            print("|-- 2.2 Read by Id")
            print("| 3. Update product")
            print("| 4. Delete product")
            print("|-- 4.1 Delete specific product")
            print("|-- 4.2 Delete all products")
            print("| 5. Sell product")
            print("| 6. Purchase product")
            print("| 0. Exit")
            options = int(input("\nyour input: "))
        except:
            print("Invalid option")
            continue
```

รูปภาพที่ 4- 7 ฟังก์ชัน product_handler

4.3.2 ฟังก์ชัน sale_handler

ฟังก์ชัน sale_handler ใช้สำหรับบันทึกและจัดการข้อมูลการขายสินค้า โดยจะอ้างอิงรหัสสินค้า (Product ID) ที่มีอยู่ในคลังเพื่อบันทึกยอดขายใหม่ ข้อมูลจะถูกจัดเก็บในไฟล์ sales.bin ตามรูปแบบที่กำหนดใน sale_fmt โครงสร้างข้อมูล (sale_fmt = "<6s6sifdd") ประกอบด้วย

1) รหัสการขาย (SaleDetail ID)
2) รหัสสินค้า (Product ID)
3) จำนวนที่ขาย (Quantity)
4) วันที่รับขาย (Created_at)
5) วันที่อัพเดตล่าสุด (Updated_at)

เมื่อมีการขายสินค้า ฟังก์ชันนี้จะช่วยให้สามารถบันทึกข้อมูลลงไฟล์เพื่อใช้ตรวจสอบย้อนหลังได้ และยังสามารถนำไปใช้ในการสรุปรายงานภายหลัง

```
def sale_handler(path, fmt, size):
    while True:
        try:
            print("\nPlease choose from list:")
            print("\n| 1. Read sale details")
            print("|-- 1.1 Read all fields")
            print("|-- 1.2 Read by Id")
            print("| 2. Update sale details by Id")
            print("| 3. Delete sale details")
            print("|-- 3.1 Delete specific sale")
            print("|-- 3.2 Delete all sales")
            print("| 0. Exit")
            options = int(input("\nyour input: "))
        except:
            print("Invalid option")
            continue
```

รูปภาพที่ 4- 8 ฟังก์ชัน sale_handler

4.3.3 ฟังก์ชัน purchase_handler

ฟังก์ชัน purchase_handler ใช้สำหรับบันทึกข้อมูลการซื้อสินค้าเข้าคลัง เพื่ออัปเดตจำนวนสินค้าให้ทันสมัยอยู่เสมอ โดยข้อมูลการซื้อจะถูกเก็บลงไฟล์ purchases.bin ตามโครงสร้าง purchase_fmt โครงสร้างข้อมูล (purchase_fmt = "<6s6sif255sdd") ประกอบด้วย

1) รหัสการซื้อ (Purchase ID)
2) รหัสสินค้า (Product ID)
3) จำนวนที่สั่งซื้อ (Quantity)
4) ยอดรวม (Total)
5) หมายเหตุ (Note)
6) วันที่สั่งซื้อ (Create_at)
7) วันที่อัพเดตล่าสุด (Updated_at)

ฟังก์ชันนี้ทำให้สามารถเก็บประวัติการสั่งซื้อสินค้าเพื่ออ้างอิงภายหลัง และใช้ในการตรวจสอบการหมุนเวียนของสินค้าภายในคลัง

```
def purchase_handler(path, fmt, size):
    while True:
        try:
            print("\nPlease choose from list:")
            print("\n| 1. Read purchase details")
            print("|-- 1.1 Read all fields")
            print("|-- 1.2 Read by Id")
            print("| 2. Update purchase details by Id")
            print("| 3. Delete purchase details")
            print("|-- 3.1 Delete specific purchase")
            print("|-- 3.2 Delete all purchases")
            print("| 0. Exit")
            options = int(input("\nyour input: "))
        except:
            print("Invalid option")
            continue
```

รูปภาพที่ 4- 9 ฟังก์ชัน purchase_handler

4.3.4 ฟังก์ชัน print_report

ฟังก์ชัน print_report ทำหน้าที่สร้างรายงานสรุปข้อมูลการจัดการคลังสินค้า โดยดึงข้อมูลจากไฟล์ไบนารี ได้แก่ products.bin, sales.bin, และ purchases.bin มาประมวลผล แล้วเขียนรายงานออกมาเป็นไฟล์ข้อความ report.txt

รายงานนี้ช่วยให้ผู้ใช้งานตรวจสอบข้อมูลย้อนหลังได้ง่ายขึ้น และเป็นเครื่องมือสำคัญในการบริหารจัดการสต็อกสินค้า

```
                                    ===================================
                                        STOCK MANAGEMENT REPORT
                                    ===================================

Generated At : 2025-10-02 00:48:10 (+07:00)
App Version : 1.0
Endianness : Little-Endian
Encoding : UTF-8 (fixed-length)

[1] PRODUCT SUMMARY
+--------+-----------------------------+-------------+----------+------------+-------------+---------------+---------------+-----------+
| SellId | Name                        | Category    | Price    | Sell Amount| Prod Remain | Last Sell     | Last Purchase | Status    |
+--------+-----------------------------+-------------+----------+------------+-------------+---------------+---------------+-----------+
| S00001 | น้ำดื่มสิงห์ 600 มล.          | เครื่องดื่ม  |     7.00 |         20 |          60 |   24/09 20:50 |         24/09 | Active    |
| S00002 | ขนมปังเลอแปงไส้ครีม           | ขนม          |    14.00 |         10 |          45 |   01/10 13:24 |         01/10 | Restock   |
| S00003 | น้ำดื่มสิงห์ 600 มล.          | เครื่องดื่ม  |     7.00 |         25 |          60 |   24/09 20:50 |         24/09 | Active    |
| S00004 | ขนมปังเลอแปงไส้ครีม           | ขนม          |    14.00 |          7 |          45 |   01/10 13:24 |         01/10 | Restock   |
+--------+-----------------------------+-------------+----------+------------+-------------+---------------+---------------+-----------+
```

รูปภาพที่ 4- 10 การแสดงผลจากฟังก์ชัน print_report

4.4 ฟังก์ชันเมนูระบบจัดการข้อมูลสินค้า

4.4.1 ฟังก์ชันเพิ่มสินค้า (case 1) มีหน้าที่ในการรับข้อมูลสินค้าใหม่จากผู้ใช้แล้วบันทึกลงไฟล์ฐานข้อมูล โดยเริ่มจากการตรวจสอบและจัดรูปแบบรหัสสินค้า (Product ID) ให้ถูกต้อง ต้องขึ้นต้นด้วย "P" และมีความยาวไม่เกิน 6 ตัวอักษร พร้อมทั้งตรวจสอบว่าไม่มีการซ้ำกับสินค้าที่มีอยู่แล้วในไฟล์ จากนั้นรับข้อมูลรายละเอียดสินค้า เช่น ชื่อสินค้า (Product Name) ซึ่งต้องไม่ว่างหรือยาวเกิน 255 ตัวอักษร ประเภทสินค้า (Category) ที่เลือกจากเมนูที่กำหนด ปริมาณสินค้า (Quantity) ที่ต้องไม่เป็นค่าติดลบ และหน่วยสินค้า (Unit) ที่เลือกจากตัวเลือกที่มีอยู่ เช่น ชิ้น ขวด หรือกล่อง นอกจากนี้ยังมีการกรอกราคาขาย (Selling Price) โดยต้องไม่เป็นค่าติดลบ สุดท้ายโปรแกรมจะกำหนดสถานะสินค้า (Status) โดยอิงจากจำนวนสินค้าคงเหลือ เช่น Active, Restock หรือ Deactive เพื่อช่วยในการจัดการสต็อกสินค้าอย่างมีประสิทธิภาพ ข้อมูลที่ได้จะถูกนำไปบันทึกในไฟล์ไบนารีเพื่อใช้งานต่อไปในระบบ

```python
case 1:
    try:
        # Get and validate product ID
        while True:
            product_id = input("Enter product ID (start with P): ").strip()
            if not product_id:
                print("Error: Product ID cannot be empty")
                continue
            if len(product_id) > 6:
                print("Error: Product ID cannot be longer than 6 characters")
                continue
            # Format product ID
            product_id = "0" * (5 - len(product_id)) + product_id
            if not product_id.startswith("P"):
                product_id = "P" + product_id
            break

        # Check if product ID already exists
        if os.path.exists(path):
            with open(path, "rb") as check_file:
                while chuck := check_file.read(size):
                    pId, *_ = st.unpack(fmt, chuck)
                    if decode_str(pId) == product_id:
                        print(f"Error: Product ID {product_id} already exists!")
                        return
```

รูปภาพที่ 4- 11 การกำหนดรหัสสินค้า

```python
# Get and validate product name
while True:
    name = input("Enter product name: ").strip()
    if not name:
        print("Error: Product name cannot be empty")
        continue
    if len(name) > 255:
        print("Error: Product name is too long (max 255 characters)")
        continue
    break
```

รูปภาพที่ 4- 12 การกำหนดชื่อสินค้า

```python
# Get and validate category
print("\nPlease choose category from list: ")
print("\n1. เครื่องดื่ม\n2. ขนม\n3. อาหารแห้ง\n4. ของสด\n5. เครื่องปรุง")
while True:
    try:
        cat_choice = int(input("\nEnter Category (1-5): "))
        if 1 <= cat_choice <= 5:
            category = category_list(cat_choice)
            if category:
                break
        print("Error: Please select a valid category (1-5)")
    except ValueError:
        print("Error: Please enter a valid number")

# Get and validate quantity
while True:
    try:
        quantity = int(input("Enter quantity: "))
        if quantity < 0:
            print("Error: Quantity cannot be negative")
            continue
        break
    except ValueError:
        print("Error: Please enter a valid number")
```

รูปภาพที่ 4- 13 การกำหนดประเภทสินค้า

```python
# Get and validate unit
print("\nPlease choose unit from list: ")
print("\n1. ชิ้น\n2. ขวด\n3. แพ็ค\n4. กล่อง\n5. ห่อ")
while True:
    try:
        unit_choice = int(input("\nEnter Unit (1-5): "))
        if 1 <= unit_choice <= 5:
            unit = unit_list(unit_choice)
            if unit:
                break
        print("Error: Please select a valid unit (1-5)")
    except ValueError:
        print("Error: Please enter a valid number")
```

รูปภาพที่ 4- 14 การกำหนดหน่วยนับของสินค้า

```python
# Get and validate selling price
while True:
    try:
        sell_price = float(input("Enter sell price: "))
        if sell_price < 0:
            print("Error: Price cannot be negative")
            continue
        break
    except ValueError:
        print("Error: Please enter a valid price")

# Determine status based on quantity
status = "Active" if quantity >= 50 else "Restock" if quantity >= 20 else "Deactive"
```

รูปภาพที่ 4- 15 การกำหนดราคาขายและสถานะ

4.4.2 ฟังก์ชันการแสดงข้อมูลสินค้า (case 2)

ฟังก์ชันในส่วนนี้มีหน้าที่หลักคือ แสดงข้อมูลสินค้า ที่บันทึกไว้ในฐานข้อมูล โดยผู้ใช้สามารถเลือกได้ว่าจะอ่านข้อมูลสินค้าทั้งหมด หรือค้นหาเฉพาะสินค้าตามรหัสสินค้า (Product ID) เริ่มต้นโปรแกรมจะตรวจสอบก่อนว่ามีไฟล์ฐานข้อมูลอยู่หรือไม่ หากไม่มีจะแจ้งว่าไม่พบฐานข้อมูลสินค้า จากนั้นจะแสดงเมนูให้เลือก 2 แบบคือ (1) แสดงสินค้าทั้งหมด และ (2) แสดงเฉพาะสินค้าที่ต้องการค้นหา

ถ้าเลือก แสดงสินค้าทั้งหมด (Read all products) โปรแกรมจะเปิดไฟล์ฐานข้อมูลสินค้าแบบ Binary แล้วอ่านข้อมูลแต่ละเรคอร์ดมาถอดรหัส (unpack) จากนั้นเรียกใช้ฟังก์ชัน print_product_details เพื่อจัดรูปแบบการแสดงผลให้อ่านง่ายในลักษณะเป็นตาราง หากไม่พบสินค้าเลยก็จะแจ้งว่าไม่มีข้อมูลในฐานข้อมูล

ถ้าเลือก ค้นหาด้วย Product ID โปรแกรมจะให้กรอกรหัสสินค้า ซึ่งต้องขึ้นต้นด้วย “P” และมีความยาวไม่เกิน 6 ตัวอักษร จากนั้นโปรแกรมจะค้นหาภายในไฟล์ว่ามีรหัสสินค้านี้หรือไม่ ถ้าพบจะแสดงรายละเอียดสินค้านั้นแบบจัดรูปแบบ เช่น ชื่อสินค้า หมวดหมู่ จำนวน หน่วย ราคา และสถานะ หากไม่พบก็จะแจ้งข้อความว่าไม่มีสินค้านั้นอยู่ในระบบ

```
case 2:
    def print_product_details(product_data):
        """Helper function to print product details in a formatted way"""
        pId, name, category, quantity, unit, sell_price, status = product_data
        print(f"| {decode_str(pId):<14}{decode_str(name):<40}{decode_str(category):<12}{


    try:
        if not os.path.exists(path):
            print("No product database found.")
            return

        print("\nRead Options:")
        print("1. Read all products")
        print("2. Read specific product by ID")

        while True:
            try:
                read_opt = int(input("\nEnter option (1-2): "))
                if read_opt in [1, 2]:
                    break
                print("Error: Please select a valid option (1 or 2)")
            except ValueError:
                print("Error: Please enter a valid number")
```

รูปภาพที่ 4- 16 ฟังก์ชันแสดงข้อมูลสินค้า

```
if read_opt == 1:
    # Read all products
    products_found = False
    print("\nProduct List:")
    with open(path, "rb") as file:
        print("="*150)
        # header
        print("| Product ID | Name    | Category    | Quantity    | Unit    | Price       | Status       |")
        print("="*150)
        while chuck := file.read(size):
            products_found = True
            print_product_details(st.unpack(fmt, chuck))
        print("="*150)

    if not products_found:
        print("No products found in database.")
```

รูปภาพที่ 4- 17 การแสดงสินค้าทั้งหมด

```python
elif read_opt == 2:
    # Read specific product
    while True:
        productId = input("Enter product ID (start with P): ").strip()
        if not productId:
            print("Error: Product ID cannot be empty")
            continue
        if len(productId) > 6:
            print("Error: Product ID cannot be longer than 6 characters")
            continue
        productId = "0" * (5 - len(productId)) + productId
        if not productId.startswith("P"):
            productId = "P" + productId
        break

    product_found = False
    with open(path, "rb") as file:
        while chuck := file.read(size):
            unpacked_data = st.unpack(fmt, chuck)
            if decode_str(unpacked_data[0]) == productId:
                print(f"\nProduct {productId} found:")
                print("="*60)
                print(f"| Product ID : {decode_str(unpacked_data[0])}")
                print(f"| Name       : {decode_str(unpacked_data[1])}")
                print(f"| Category   : {decode_str(unpacked_data[2])}")
                print(f"| Quantity   : {unpacked_data[3]}")
                print(f"| Unit       : {decode_str(unpacked_data[4])}")
                print(f"| Price      : {unpacked_data[5]:.2f}")
                print(f"| Status     : {decode_str(unpacked_data[6])}")
                print("="*60)
                product_found = True
                break
```

รูปภาพที่ 4- 18 การแสดงสินค้าจากการค้นหาด้วยรหัสสินค้า

4.4.3 ฟังก์ชันการแก้ไขข้อมูลสินค้า (case 3)

ฟังก์ชันในส่วนนี้มีหน้าที่หลักคือ ให้ผู้ใช้สามารถ แก้ไขข้อมูลสินค้าที่มีอยู่แล้วในระบบ โดยเริ่มจากให้กรอกรหัสสินค้า (Product ID) ซึ่งต้องมีรูปแบบถูกต้องตามข้อกำหนด จากนั้นโปรแกรมจะเปิดไฟล์ฐานข้อมูลสินค้าแบบ Binary และค้นหาสินค้าที่ตรงกับรหัสที่ผู้ใช้กรอก หากไม่พบสินค้าจะแจ้งข้อความว่าไม่มีสินค้านั้นในระบบ

เมื่อพบสินค้าตัวนั้น โปรแกรมจะแสดงค่าปัจจุบันของสินค้าทั้งหมด เช่น ชื่อสินค้า หมวดหมู่ จำนวนหน่วย และราคาขาย พร้อมให้ผู้ใช้เลือกว่าจะ แก้ไขฟิลด์ใดบ้าง ตั้งแต่ 1–5 หรือแก้ไขทุกฟิลด์ (6) หากผู้ใช้เลือกยกเลิกจะหยุดกระบวนการทันที

สำหรับแต่ละฟิลด์ โปรแกรมจะมีการตรวจสอบความถูกต้องของข้อมูล เช่น ตรวจสอบชื่อไม่ว่างหรือไม่เกิน 255 ตัวอักษร ตรวจสอบจำนวนสินค้าว่าต้องไม่เป็นลบ ตรวจสอบราคาขายต้องเป็นตัวเลขบวก และให้เลือกหมวดหมู่หรือหน่วยจากรายการที่กำหนดไว้ล่วงหน้า

หลังจากผู้ใช้กรอกข้อมูลใหม่ทั้งหมด ระบบจะคำนวณ สถานะของสินค้า (Status) โดยอิงจากจำนวนสินค้าในสต็อก เช่น จำนวนมากกว่า 50 เป็น “Active”, 20–49 เป็น “Restock” และต่ำกว่า 20 เป็น “Deactive” จากนั้นสร้างเรคอร์ดสินค้าใหม่แบบ Binary และเขียนทับเรคอร์ดเดิมในไฟล์ฐานข้อมูล โดยใช้การอัปเดตแบบปลอดภัย เพื่อป้องกันข้อมูลเสียหาย

สุดท้ายโปรแกรมจะแจ้งผลสำเร็จของการแก้ไข พร้อมแสดงค่าปัจจุบันทั้งหมดของสินค้า ทำให้ผู้ใช้สามารถจัดการข้อมูลสินค้าได้อย่างครบถ้วน ถูกต้อง และสะดวกต่อการดูแลสต็อกสินค้า

```
case 3:
    def print_current_values(product_data):
        """Helper function to print current product values"""
        pId, name, category, quantity, unit, sell_price, status = product_data
        print("\nCurrent Product Values:")
        print("+" + "-" * 50 + "+")
        print(f"| 1. Name     : {decode_str(name):<39} |")
        print(f"| 2. Category : {decode_str(category):<39} |")
        print(f"| 3. Quantity : {quantity:<39} |")
        print(f"| 4. Unit     : {decode_str(unit):<39} |")
        print(f"| 5. Price    : {sell_price:<39.2f} |")
        print("+" + "-" * 50 + "+")

    try:
        if not os.path.exists(path):
            print("No product database found.")
            return
```

รูปภาพที่ 4- 19 ฟังก์ชันการแก้ไขข้อมูลสินค้า

```python
# Get and validate product ID
while True:
    productId = input("Enter product ID (start with P): ").strip()
    if not productId:
        print("Error: Product ID cannot be empty")
        continue
    if len(productId) > 6:
        print("Error: Product ID cannot be longer than 6 characters")
        continue
    productId = "0" * (5 - len(productId)) + productId
    if not productId.startswith("P"):
        productId = "P" + productId
    break
```

รูปภาพที่ 4- 20 การกรอกรหัสสินค้าที่ต้องการแก้ไข

```python
print_current_values(product_data)
print("\nUpdate Options:")
print("1. Name")
print("2. Category")
print("3. Quantity")
print("4. Unit")
print("5. Sell Price")
print("6. Update all fields")
print("0. Cancel update")

# Get valid update choice
while True:
    try:
        update_choice = int(input("\nEnter field number to update (0-6): "))
        if 0 <= update_choice <= 6:
            break
        print("Error: Please enter a valid option (0-6)")
    except ValueError:
        print("Error: Please enter a valid number")

if update_choice == 0:
    print("Update cancelled.")
    return
```

รูปภาพที่ 4- 21 การระบุฟิลด์ที่ต้องการแก้ไข

```python
    # Update selected fields
    if update_choice == 1 or update_choice == 6:
        while True:
            new_name = input("Enter new name: ").strip()
            if not new_name:
                print("Error: Name cannot be empty")
                continue
            if len(new_name) > 255:
                print("Error: Name is too long (max 255 characters)")
                continue
            break

    if update_choice == 2 or update_choice == 6:
        print("\nPlease choose category from list:")
        print("1. เครื่องดื่ม\n2. ขนม\n3. อาหารแห้ง\n4. ของสด\n5. เครื่องปรุง")
        while True:
            try:
                cat_choice = int(input("\nEnter Category (1-5): "))
                if 1 <= cat_choice <= 5:
                    new_category = category_list(cat_choice)
                    if new_category:
                        break
                print("Error: Please select a valid category (1-5)")
            except ValueError:
                print("Error: Please enter a valid number")
```

รูปภาพที่ 4- 22 การแก้ไขชื่อและการแก้ไขประเภท

```python
if update_choice == 3 or update_choice == 6:
    while True:
        try:
            new_quantity = int(input("Enter new quantity: "))
            if new_quantity < 0:
                print("Error: Quantity cannot be negative")
                continue
            break
        except ValueError:
            print("Error: Please enter a valid number")

if update_choice == 4 or update_choice == 6:
    print("\nPlease choose unit from list:")
    print("1. ชิ้น\n2. ขวด\n3. แพ็ค\n4. กล่อง\n5. ห่อ")
    while True:
        try:
            unit_choice = int(input("\nEnter Unit (1-5): "))
            if 1 <= unit_choice <= 5:
                new_unit = unit_list(unit_choice)
                if new_unit:
                    break
            print("Error: Please select a valid unit (1-5)")
        except ValueError:
            print("Error: Please enter a valid number")
```

รูปภาพที่ 4- 23 การแก้ไขจำนวนและการแก้ไขหน่วยนับ

```python
if update_choice == 5 or update_choice == 6:
    while True:
        try:
            new_sell_price = float(input("Enter new sell price: "))
            if new_sell_price < 0:
                print("Error: Price cannot be negative")
                continue
            break
        except ValueError:
            print("Error: Please enter a valid price")

# Update status based on new quantity
new_status = "Active" if new_quantity >= 50 else "Restock" if new_quantity >= 20 else "Deactive"
```

รูปภาพที่ 4- 24 การแก้ไขจำนวนและการแก้ไขหน่วยนับ

4.4.4 ฟังก์ชันการลบข้อมูลสินค้า (case 4)

ฟังก์ชันในส่วนนี้มีหน้าที่หลักคือ ลบข้อมูลสินค้าออกจากระบบ ทั้งแบบลบทีละรายการหรือแบบลบทั้งหมด โดยเริ่มจากตรวจสอบว่ามีไฟล์ฐานข้อมูลสินค้าหรือไม่ หากไม่มีจะแจ้งเตือนผู้ใช้ทันที

ผู้ใช้จะได้รับตัวเลือกว่าจะลบสินค้ารายการเดียว , ลบสินค้าทั้งหมด หรือยกเลิกการลบ สำหรับการลบสินค้ารายตัว โปรแกรมจะให้กรอกรหัสสินค้า (Product ID) และตรวจสอบความถูกต้องของรหัส จากนั้นค้นหาในฐานข้อมูล หากไม่พบจะแจ้งว่าไม่มีสินค้านั้น

ก่อนการลบจริง โปรแกรมจะแสดง รายละเอียดสินค้าที่ต้องการลบ เพื่อให้ผู้ใช้ตรวจสอบ และมีฟังก์ชันยืนยันการลบ (confirm_delete) ซึ่งผู้ใช้ต้องตอบ yes หรือ no เพื่อป้องกันการลบโดยไม่ตั้งใจ

หากผู้ใช้ยืนยันการลบ ระบบจะสร้างไฟล์ชั่วคราวแล้วคัดลอกเรคอร์ดทั้งหมด ยกเว้นสินค้าที่จะลบลงไป จากนั้นแทนที่ไฟล์ฐานข้อมูลเดิมด้วยไฟล์ชั่วคราว ทำให้การลบปลอดภัยและไม่ทำลายข้อมูลอื่น

ในกรณีที่เลือกลบทั้งหมด ระบบจะถามยืนยันอีกครั้งเพราะการลบนี้ ไม่สามารถย้อนกลับได้ หากยืนยันจะลบไฟล์ฐานข้อมูลทั้งหมดออกไปทันที นอกจากนี้ยังมีการจัดการข้อผิดพลาด เช่น ไฟล์ไม่พบหรือไม่มีสิทธิ์เข้าถึงไฟล์ เพื่อให้โปรแกรมทำงานได้อย่างราบรื่นและปลอดภัยต่อข้อมูล

```python
case 4:
    def confirm_delete(message):
        """Helper function to confirm deletion"""
        while True:
            confirm = input(f"{message} (yes/no): ").lower().strip()
            if confirm in ['yes', 'y']:
                return True
            if confirm in ['no', 'n']:
                return False
            print("Please answer 'yes' or 'no'")

    try:
        if not os.path.exists(path):
            print("No product database found.")
            return

        print("\nDelete Options:")
        print("1. Delete specific product")
        print("2. Delete all products")
        print("0. Cancel")
```

รูปภาพที่ 4- 25 ฟังก์ชันการลบข้อมูลสินค้า

```python
elif opt == 1:
    # Delete specific product
    while True:
        productId = input("Enter product ID (start with P): ").strip()
        if not productId:
            print("Error: Product ID cannot be empty")
            continue
        if len(productId) > 6:
            print("Error: Product ID cannot be longer than 6 characters")
            continue
        productId = "0" * (5 - len(productId)) + productId
        if not productId.startswith("P"):
            productId = "P" + productId
        break

    # First check if product exists and show its details
    product_found = False
    product_details = None
    with open(path, "rb") as file:
        while chuck := file.read(size):
            data = st.unpack(fmt, chuck)
            if decode_str(data[0]) == productId:
                product_found = True
                product_details = data
                break

    if not product_found:
        print(f"\nProduct {productId} not found in database.")
        return
```

รูปภาพที่ 4- 26 การลบข้อมูลสินค้ารายการเดียว

```python
elif opt == 2:
    # Delete all products
    if not confirm_delete("\nAre you sure you want to delete ALL products? This action cannot be undone!"):
        print("Delete operation cancelled.")
        return

    try:
        os.remove(path)
        print("\nAll products have been deleted successfully.")
    except FileNotFoundError:
        print("No product database found.")
    except PermissionError:
        print("Error: Permission denied. Please check file permissions.")
```

รูปภาพที่ 4- 27 การลบข้อมูลสินค้าทั้งหมด

4.4.5 ฟังก์ชันการขายสินค้าและอัปเดตสต็อก (case 5)

ฟังก์ชันในส่วนนี้มีหน้าที่หลักคือ บันทึกการขายสินค้า และอัปเดตจำนวนสินค้าคงเหลือในคลังได้อย่างอัตโนมัติ โดยเริ่มจากการให้ผู้ใช้กรอกรหัสสินค้า (Product ID) และตรวจสอบความถูกต้องของรหัส หากรหัสสั้นเกินไปหรือไม่ขึ้นต้นด้วย “P” ระบบจะปรับให้เป็นฟอร์แมตมาตรฐาน

โปรแกรมจะค้นหาสินค้าในฐานข้อมูล หากไม่พบจะแจ้งข้อผิดพลาด หากพบก็จะแสดง รายละเอียดสินค้า ทั้งชื่อ หมวดหมู่ จำนวน หน่วย ราคา และสถานะปัจจุบันของสินค้า

ผู้ใช้สามารถระบุจำนวนสินค้าที่ต้องการขาย โปรแกรมจะตรวจสอบว่ามีจำนวนเพียงพอ หากจำนวนที่ขายมากกว่าสต็อก จะไม่อนุญาตให้ขาย

เมื่อขายสำเร็จ โปรแกรมจะ สร้างรหัสขายอัตโนมัติ (Sale ID) โดยเช็กจากไฟล์บันทึกการขายเก่า หากมีการขายเดิม จะเพิ่มหมายเลขต่อเนื่อง และบันทึกเวลาปัจจุบันเพื่อใช้ติดตามการขาย

ต่อมา โปรแกรมจะบันทึกข้อมูลการขายลงไฟล์สรุปการขาย หากมีการขายเดิมของสินค้านั้น โปรแกรมจะรวมจำนวนและคำนวณยอดรวมใหม่ ส่วนสต็อกสินค้าหลักจะถูก ลดจำนวนตามที่ขายไป และอัปเดตสถานะสินค้าเป็น Active / Restock / Deactive ตามปริมาณที่เหลือ และสุดท้ายโปรแกรมเขียนข้อมูลใหม่ลงไฟล์สินค้าหลัก ทำให้ สต็อกสินค้าและบันทึกการขาย เป็นปัจจุบัน และแจ้งผลการขายให้ผู้ใช้ทราบ

```
case 5:
    product_id = input("Enter product ID (start with P): ")
    if "0" not in product_id or len(product_id) < 6:
        product_id = "0" * (5 - len(product_id)) + product_id
        if not product_id.startswith("P"):
            product_id = "P" + product_id
    try:
        index = 0
        found_product = None
        with open(path, "rb") as file:
            while chuck := file.read(size):
                current_product = st.unpack(fmt, chuck)
                if decode_str(current_product[0]) == product_id:
                    found_product = current_product
                    break
                index += 1

        if not found_product:
            print(f"Error: Product ID {product_id} not found!")
            continue
```

รูปภาพที่ 4- 28 ฟังก์ชันการขายสินค้าและอัปเดตสต็อก

```
product_id = input("Enter product ID (start with P): ")
if "0" not in product_id or len(product_id) < 6:
    product_id = "0" * (5 - len(product_id)) + product_id
    if not product_id.startswith("P"):
        product_id = "P" + product_id
try:
    index = 0
    found_product = None
    with open(path, "rb") as file:
        while chuck := file.read(size):
            current_product = st.unpack(fmt, chuck)
            if decode_str(current_product[0]) == product_id:
                found_product = current_product
                break
            index += 1

    if not found_product:
        print(f"Error: Product ID {product_id} not found!")
        continue

    pId, name, category, quantity, unit, sell_price, status = found_product

    print(f"\n| productId: {decode_str(pId)}")
    print(f"| Name: {decode_str(name)}")
    print(f"| Category: {decode_str(category)}")
    print(f"| Quantity: {quantity}")
    print(f"| Unit: {decode_str(unit)}")
    print(f"| Sell_price: {sell_price}")
    print(f"| Status: {decode_str(status)}")

    sell = int(input("\nEnter quantity to sell: "))
    if sell > quantity:
        print("Error: Not enough stock to sell.")
        continue
```

รูปภาพที่ 4- 29 การระบุสินค้าและจำนวนที่ขาย

4.4.6 ฟังก์ชันการการสั่งซื้อสินค้าและอัปเดตสต็อก (case 6)

ฟังก์ชันในส่วนนี้มีหน้าที่หลักคือ เพิ่มจำนวนสินค้าคงคลังผ่านการซื้อสินค้า และบันทึกการสั่งซื้อให้อัตโนมัติ โดยเริ่มจากให้ผู้ใช้กรอกรหัสสินค้า (Product ID) และตรวจสอบความถูกต้องของรหัส หากรหัสสั้นเกินไปหรือไม่ขึ้นต้นด้วย “P” ระบบจะปรับให้อยู่ในฟอร์แมตมาตรฐาน

โปรแกรมจะค้นหาสินค้าในฐานข้อมูล หากพบก็จะแสดง รายละเอียดสินค้า ปัจจุบัน เช่น ชื่อ หมวดหมู่ จำนวน หน่วย ราคา และสถานะ เพื่อให้ผู้ใช้ตรวจสอบก่อนทำการสั่งซื้อ

ผู้ใช้ระบุจำนวนสินค้าที่ต้องการซื้อและคำอธิบายของการสั่งซื้อ โปรแกรมจะ สร้างรหัสสั่งซื้ออัตโนมัติ (Purchase ID) โดยเช็กไฟล์บันทึกการสั่งซื้อเก่าและเพิ่มหมายเลขต่อเนื่อง หากมีการซื้อเดิมของสินค้านั้น ระบบจะรวมจำนวนและคำนวณยอดรวมใหม่

หลังจากนั้น โปรแกรมจะบันทึกข้อมูลการสั่งซื้อลงไฟล์สั่งซื้อ และ อัปเดตจำนวนสินค้าในคลัง โดยเพิ่มจำนวนตามที่สั่งซื้อ พร้อมปรับสถานะสินค้าเป็น Active / Restock / Deactive ตามจำนวนใหม่

สุดท้าย โปรแกรมเขียนข้อมูลสินค้าหลักลงไฟล์ ทำให้สต็อกและบันทึกการสั่งซื้อ เป็นปัจจุบันอย่างครบวงจร และแจ้งผลให้ผู้ใช้ทราบว่าได้สร้างหรืออัปเดตการสั่งซื้อเรียบร้อย

```
case 6:
    product_id = input("Enter product ID (start with P): ")
    if "0" not in product_id or len(product_id) < 6:
        product_id = "0" * (5 - len(product_id)) + product_id
        if not product_id.startswith("P"):
            product_id = "P" + product_id
    try:
        index = 0
        found_product = None
        with open(path, "rb") as file:
            while chuck := file.read(size):
                current_product = st.unpack(fmt, chuck)
                if decode_str(current_product[0]) == product_id:
                    found_product = current_product
                    break
                index += 1

        if not found_product:
            print(f"Error: Product ID {product_id} not found!")
            continue
```

รูปภาพที่ 4- 30 ฟังก์ชันการสั่งซื้อและอัปเดตสต็อก

```python
if not found_product:
    print(f"Error: Product ID {product_id} not found!")
    continue

pId, name, category, quantity, unit, sell_price, status = found_product
print(f"\n| Name: {decode_str(name)}")
print(f"| Category: {decode_str(category)}")
print(f"| Quantity: {quantity}")
print(f"| Unit: {decode_str(unit)}")
print(f"| Sell_price: {sell_price}")
print(f"| Status: {decode_str(status)}")

# Auto-generate Purchase ID
next_purchase_num = 1
if os.path.exists(purchase_path):
    with open(purchase_path, "rb") as purchase_file:
        while chunk := purchase_file.read(purchase_size):
            purId, *_ = st.unpack(purchase_fmt, chunk)
            purId_decoded = decode_str(purId)
            if purId_decoded.startswith("I") and purId_decoded[1:].isdigit():
                num = int(purId_decoded[1:])
                if num >= next_purchase_num:
                    next_purchase_num = num + 1
purchase_id = f"I{next_purchase_num:05d}"

purchase_qty = int(input("Enter quantity to purchase: "))
if purchase_qty <= 0:
    print("Error: Quantity must be greater than 0")
    return

purchase_desc = input("Enter purchase description: ")
current_time = datetime.datetime.now().timestamp()
new_sell_price = sell_price - 2
total = purchase_qty * new_sell_price
```

รูปภาพที่ 4- 31 การระบุสินค้าและจำนวนที่สั่งซื้อ

4.5 ฟังก์ชันเมนูระบบจัดการข้อมูลการขาย

4.5.1 ฟังก์ชันแสดงข้อมูลการขาย (case 1)

ฟังก์ชันส่วนนี้มีหน้าที่หลักคือแสดงข้อมูลการขายที่บันทึกไว้ในไฟล์ไบนารี ผู้ใช้สามารถเลือกอ่านทั้งหมด หรือ ค้นหาตาม Sale ID ถ้าเลือกอ่าน ทั้งหมด โปรแกรมจะเปิดไฟล์ไบนารี, อ่านแต่ละเรคคอร์ด, ถอดรหัส (unpack) แล้วจัดรูปแบบแสดงผลเป็นตารางพร้อมหัวข้อ เช่น Sale ID, Product ID, Quantity, Total Amount, Created At, Updated At

ถ้าเลือกอ่าน เฉพาะ Sale ID โปรแกรมจะให้ผู้ใช้กรอกรหัสการขาย (ต้องขึ้นต้นด้วย “S” และไม่เกิน 6 ตัวอักษร) จากนั้นค้นหาในไฟล์และแสดงรายละเอียดแบบจัดรูปแบบ หากไม่พบจะแจ้งข้อความว่าไม่มีเรคคอร์ดนี้

```
def sale_handler(path, fmt, size):
    while True:
        try:
            print("\nPlease choose from list:")
            print("\n| 1. Read sale details")
            print("|-- 1.1 Read all fields")
            print("|-- 1.2 Read by Id")
            print("| 2. Update sale details by Id")
            print("| 3. Delete sale details")
            print("|-- 3.1 Delete specific sale")
            print("|-- 3.2 Delete all sales")
            print("| 0. Exit")
            options = int(input("\nyour input: "))
        except:
            print("Invalid option")
            continue
```

รูปภาพที่ 4- 32 ฟังก์ชันเมนูระบบจัดการข้อมูลการขาย

```
if read_opt == 1:
    with open(path, "rb") as file:
        print("="*94)
        print("| Sale ID  | " + "Product ID  | " + "Quantity  | " + "Total Amount       | " + "Created At     | " + "Updated At     |")
        print("="*94)
        while chuck := file.read(size):
            saleId, productId, quantity, totalAmount, created_at, updated_at = st.unpack(fmt, chuck)
            print(f"| {decode_str(saleId):<8} | {decode_str(productId):<11} | {quantity:<9} | {totalAmount:<19} | {created_at:<14.2f} | {updated_a
        print("="*94)
```

รูปภาพที่ 4- 33 การแสดงข้อมูลการขายทั้งหมด

```
elif read_opt == 2:
    sale_id = input("\nEnter Sale ID: ")
    if "0" not in sale_id or len(sale_id) < 6:
        sale_id = "0" * (5 - len(sale_id)) + sale_id
        if not sale_id.startswith("S"):
            sale_id = "S" + sale_id
    with open(path, "rb") as file:
        while chuck := file.read(size):
            sales = st.unpack(fmt, chuck)
            if decode_str(sales[0]) == sale_id:
                print(f"Sale details for ID {sale_id}: ")
                print("="*60)
                print(f"| Sale ID      : {decode_str(sales[0])}")
                print(f"| Product ID   : {decode_str(sales[1])}")
                print(f"| Quantity     : {sales[2]}")
                print(f"| Total Amount : {sales[3]}")
                print(f"| Created At   : {sales[4]:.2f}")
                print(f"| Updated At   : {sales[5]:.2f}")
                print("="*60)
```

รูปภาพที่ 4- 34 การแสดงข้อมูลการขายเฉพาะรายการที่ต้องการค้นหา

4.5.2 ฟังก์ชันการอัปเดตข้อมูลการขายตาม Sale ID (case 2)

ฟังก์ชันส่วนนี้ช่วยให้ผู้ใช้แก้ไขข้อมูลขายที่มีอยู่โปรแกรมจะค้นหาเร็กคอร์ดที่ตรงกับ Sale ID จากนั้นจะแสดงค่าปัจจุบันของฟิลด์ทุกตัว เช่น Quantity, Total Amount, Created At, Updated At โดยผู้ใช้สามารถป้อนค่าใหม่ หรือเก็บค่าเดิมโดยกรอก -1 ข้อมูลใหม่จะถูกบันทึกลงไฟล์ชั่วคราวก่อน จากนั้นแทนที่ไฟล์เดิมเพื่อความปลอดภัย และโปรแกรมจะบอกผู้ใช้เมื่ออัปเดตเรียบร้อย หรือแจ้งว่าไม่พบ Sale ID นี้ในฐานข้อมูล

```
case 2:
    sale_id = input("Enter Sale ID to update: ").strip()
    if "0" not in sale_id or len(sale_id) < 6:
        sale_id = "0" * (5 - len(sale_id)) + sale_id
        if not sale_id.startswith("S"):
            sale_id = "S" + sale_id
    temp_path = ".tmp"
    found = False
    with open(path, "rb") as file, open(temp_path, "wb") as tmp:
        while chunk := file.read(size):
            record = st.unpack(fmt, chunk)
            if decode_str(record[0]) == sale_id:
                found = True
                print("Current values:")
                print(f"| Sale ID      : {decode_str(record[0])}")
                print(f"| Product ID   : {decode_str(record[1])}")
                print(f"| Quantity     : {record[2]}")
                print(f"| Total Amount : {record[3]}")
                print(f"| Created At   : {record[4]:.2f}")
                print(f"| Updated At   : {record[5]:.2f}")
```

รูปภาพที่ 4- 35 ฟังก์ชันการอัปเดตข้อมูลการขายตาม Sale ID

```
            new_quantity = int(input("Enter new quantity (or -1 to keep): "))
            if new_quantity == -1:
                new_quantity = record[2]
            new_total = float(input("Enter new total amount (or -1 to keep): "))
            if new_total == -1:
                new_total = record[3]
            import datetime
            updated_at = datetime.datetime.now().timestamp()
            new_record = st.pack(fmt, record[0], record[1], new_quantity, new_total, record[4], updated_at)
            tmp.write(new_record)
        else:
            tmp.write(chunk)
if found:
    os.replace(temp_path, path)
    print("Sale record updated.")
else:
    os.remove(temp_path)
    print("Sale ID not found.")
```

รูปภาพที่ 4- 36 การแก้ไขจำนวนสินค้าและยอดรวม

4.5.3 ฟังก์ชันลบข้อมูลการขาย (case 3)

ฟังก์ชันส่วนนี้มีหน้าที่ลบเร็กคอร์ดการขายที่ไม่ต้องการ ผู้ใช้สามารถเลือก ลบเฉพาะรายการ หรือ ลบทั้งหมด หากลบเฉพาะรายการ โปรแกรมจะให้กรอก Sale ID และตรวจสอบความถูกต้องของรหัส จากนั้นโปรแกรมจะสร้างไฟล์ชั่วคราวแล้วคัดลอกรายการอื่น ๆ ที่ไม่ถูกลบไปยังไฟล์ชั่วคราวก่อนแทนที่ไฟล์เดิม

หากเลือกลบทั้งหมด โปรแกรมจะให้ผู้ใช้ยืนยันก่อนเคลียร์ไฟล์ เพื่อป้องกันการลบโดยไม่ได้ตั้งใจหลังการลบ ระบบจะแจ้งผลลัพธ์ว่าเร็กคอร์ดถูกลบเรียบร้อยแล้ว หรือหากไม่พบ Sale ID จะแจ้งว่าไม่พบเร็กคอร์ด

```
case 3:
    print("Delete Options:")
    print("1. Delete specific sale")
    print("2. Delete all sales")
    print("0. Cancel")
```

รูปภาพที่ 4- 37 ฟังก์ชันลบข้อมูลการขาย

```python
if opt == 0:
    print("Cancel delete.")
elif opt == 1:
    sale_id = input("Enter Sale ID to delete: ").strip()
    if "0" not in sale_id or len(sale_id) < 6:
        sale_id = "0" * (5 - len(sale_id)) + sale_id
        if not sale_id.startswith("S"):
            sale_id = "S" + sale_id
    temp_path = ".tmp"
    found = False
    with open(path, "rb") as file, open(temp_path, "wb") as tmp:
        while chunk := file.read(size):
            record = st.unpack(fmt, chunk)
            if decode_str(record[0]) == sale_id:
                found = True
                continue
            tmp.write(chunk)
    if found:
        os.replace(temp_path, path)
        print("Sale record deleted.")
    else:
        os.remove(temp_path)
        print("Sale ID not found.")
```

รูปภาพที่ 4- 38 การลบข้อมูลการขายรายการเดียว

```python
elif opt == 2:
    confirm = input("Are you sure you want to delete all sales? (y/n): ")
    if confirm.lower() == "y":
        open(path, "wb").close()
        print("All sale records deleted.")
    else:
        print("Cancelled.")
```

รูปภาพที่ 4- 39 การลบข้อมูลการขายทั้งหมด

4.6 ฟังก์ชันเมนูระบบจัดการข้อมูลการสั่งซื้อ

4.6.1 ฟังก์ชันแสดงข้อมูลการสั่งซื้อ (case 1)

ฟังก์ชันส่วนนี้มีหน้าที่หลักคือแสดงข้อมูลการสั่งซื้อที่บันทึกไว้ในไฟล์ไบนารี ผู้ใช้สามารถเลือกอ่านทั้งหมด หรือ ค้นหาตาม Purchase ID หากเลือกอ่านทั้งหมด โปรแกรมจะเปิดไฟล์ไบนารี, อ่านแต่ละเร็กคอร์ดทีละบล็อก, ถอดรหัสข้อมูลด้วย struct.unpack และแสดงผลเป็นตารางอย่างชัดเจน

หากเลือกค้นหาตาม Purchase ID โปรแกรมจะให้กรอกรหัส โดยตรวจสอบว่าขึ้นต้นด้วย “I” และมีความยาวไม่เกิน 6 ตัวอักษร โปรแกรมจะค้นหาเร็กคอร์ดที่ตรงกับรหัส และแสดงรายละเอียด เช่น รหัสการสั่งซื้อ, รหัสสินค้า, จำนวน, ยอดรวม, หมายเหตุ, เวลาสร้างและเวลาแก้ไข หากไม่พบ Purchase ID ระบบจะแจ้งข้อความว่าไม่พบเร็กคอร์ดดังกล่าว

```
def purchase_handler(path, fmt, size):
    while True:
        try:
            print("\nPlease choose from list:")
            print("\n| 1. Read purchase details")
            print("|-- 1.1 Read all fields")
            print("|-- 1.2 Read by Id")
            print("| 2. Update purchase details by Id")
            print("| 3. Delete purchase details")
            print("|-- 3.1 Delete specific purchase")
            print("|-- 3.2 Delete all purchases")
            print("| 0. Exit")
            options = int(input("\nyour input: "))
        except:
            print("Invalid option")
            continue
```

รูปภาพที่ 4- 40 ฟังก์ชันเมนูระบบจัดการข้อมูลการสั่งซื้อ

```
case 1:
    read_opt = int(input("\n1: Read all fields\n2: Read by Id\nYour input: "))
    if not read_opt and read_opt not in [1, 2]:
        print("Invalid option")
        continue
    try:
        if read_opt == 1:
            with open(path, "rb") as file:
                print("="*110)
                print("| Purchase ID   | Product ID   | Quantity  | Total     | Note              | Created At     | Updated At
                print("="*110)
                while chuck := file.read(size):
                    purchaseId, productId, quantity, total, Note, created_at, updated_at = st.unpack(fmt, chuck)
                    print(f"| {decode_str(purchaseId):<13} | {decode_str(productId):<12} | {quantity:<9} | {total:<9} | {decode_str
                print("="*110)
```

รูปภาพที่ 4- 41 การแสดงข้อมูลการสั่งซื้อทั้งหมด

```
elif read_opt == 2:
    purchase_id = input("\nEnter Purchase ID: ")
    if "0" not in purchase_id or len(purchase_id) < 6:
        purchase_id = "0" * (5 - len(purchase_id)) + purchase_id
        if not purchase_id.startswith("I"):
            purchase_id = "I" + purchase_id
    with open(path, "rb") as file:
        while chuck := file.read(size):
            purchases = st.unpack(fmt, chuck)
            if decode_str(purchases[0]) == purchase_id:
                print(f"Purchase details for ID {purchase_id}: ")
                print("="*60)
                print(f"| Purchase ID : {decode_str(purchases[0])}")
                print(f"| Product ID  : {decode_str(purchases[1])}")
                print(f"| Quantity    : {purchases[2]}")
                print(f"| Total       : {purchases[3]}")
                print(f"| Note        : {decode_str(purchases[4])}")
                print(f"| Created At  : {purchases[5]:.2f}")
                print(f"| Updated At  : {purchases[6]:.2f}")
                print("="*60)
```

รูปภาพที่ 4- 42 การแสดงข้อมูลการสั่งซื้อรายการเดียว

4.6.2 ฟังก์ชันอัปเดตข้อมูลการสั่งซื้อ (case 2)

ฟังก์ชันนี้ช่วยให้ผู้ใช้แก้ไขข้อมูลการสั่งซื้อที่มีอยู่ ผู้ใช้กรอก Purchase ID เพื่อค้นหาเร็กคอร์ดที่ต้องการอัปเดต โปรแกรมจะแสดงค่าปัจจุบันทั้งหมดเพื่อให้ผู้ใช้ตรวจสอบก่อนแก้ไข โดยผู้ใช้สามารถแก้ไขจำนวน, ยอดรวม, และ หมายเหตุ หากไม่ต้องการแก้ไข สามารถใส่ค่า -1 หรือเว้นว่างเพื่อเก็บค่าปัจจุบันไว้ หลังแก้ไข โปรแกรมจะสร้างไฟล์ชั่วคราวและเขียนเร็กคอร์ดใหม่ พร้อมคัดลอกเร็กคอร์ดอื่น ๆ จากไฟล์เดิม และไฟล์ชั่วคราวจะถูกแทนที่ไฟล์เดิม ทำให้การอัปเดตปลอดภัย หากไม่พบ Purchase ID ระบบจะแจ้งข้อความว่าไม่พบเร็กคอร์ด

```
case 2:
    purchase_id = input("Enter Purchase ID to update: ").strip()
    if "0" not in purchase_id or len(purchase_id) < 6:
        purchase_id = "0" * (5 - len(purchase_id)) + purchase_id
        if not purchase_id.startswith("I"):
            purchase_id = "I" + purchase_id
    temp_path = ".tmp"
    found = False
    with open(path, "rb") as file, open(temp_path, "wb") as tmp:
        while chunk := file.read(size):
            record = st.unpack(fmt, chunk)
            if decode_str(record[0]) == purchase_id:
                found = True
                print("Current values:")
                print(f"| Purchase ID : {decode_str(record[0])}")
                print(f"| Product ID  : {decode_str(record[1])}")
                print(f"| Quantity    : {record[2]}")
                print(f"| Total       : {record[3]}")
                print(f"| Note        : {decode_str(record[4])}")
                print(f"| Created At  : {record[5]:.2f}")
                print(f"| Updated At  : {record[6]:.2f}")
                new_quantity = int(input("Enter new quantity (or -1 to keep): "))
                if new_quantity == -1:
                    new_quantity = record[2]
                new_total = float(input("Enter new total (or -1 to keep): "))
                if new_total == -1:
                    new_total = record[3]
                new_note = input("Enter new note (or leave blank to keep): ")
                if not new_note:
                    new_note = decode_str(record[4])
                import datetime
                updated_at = datetime.datetime.now().timestamp()
                new_record = st.pack(fmt, record[0], record[1], new_quantity, new_total, fix_str(new_note), record[5], updated_at)
                tmp.write(new_record)
```

รูปภาพที่ 4- 43 ฟังก์ชันอัปเดตข้อมูลการสั่งซื้อ

4.6.3 ฟังก์ชันลบข้อมูลการสั่งซื้อ (case 2)

ฟังก์ชันนี้ช่วยให้ผู้ใช้ลบเร็กคอร์ดการสั่งซื้อที่ไม่ต้องการ ผู้ใช้สามารถเลือก ลบเฉพาะรายการ หรือ ลบทั้งหมด หากลบเฉพาะรายการ โปรแกรมจะให้กรอก Purchase ID และตรวจสอบความถูกต้องของรหัส โปรแกรมจะสร้างไฟล์ชั่วคราวและคัดลอกเร็กคอร์ดที่ไม่ถูกลบไปยังไฟล์ชั่วคราว ก่อนแทนที่ไฟล์เดิม

หากลบทั้งหมด โปรแกรมจะให้ผู้ใช้ยืนยันก่อนลบ เพื่อป้องกันการลบโดยไม่ตั้งใจ หลังการลบ ระบบจะแจ้งผลลัพธ์ว่าเร็กคอร์ดถูกลบเรียบร้อย หรือหากไม่พบ Purchase ID จะแจ้งว่าไม่พบเร็กคอร์ด

```
case 3:
    print("Delete Options:")
    print("1. Delete specific purchase")
    print("2. Delete all purchases")
    print("0. Cancel")
    opt = int(input("Your input: "))
    if opt == 0:
        print("Cancel delete.")
```

รูปภาพที่ 4- 44 ฟังก์ชันลบข้อมูลการสั่งซื้อ

```
elif opt == 1:
    purchase_id = input("Enter Purchase ID to delete: ").strip()
    if "0" not in purchase_id or len(purchase_id) < 6:
        purchase_id = "0" * (5 - len(purchase_id)) + purchase_id
        if not purchase_id.startswith("I"):
            purchase_id = "I" + purchase_id
    temp_path = ".tmp"
    found = False
    with open(path, "rb") as file, open(temp_path, "wb") as tmp:
        while chunk := file.read(size):
            record = st.unpack(fmt, chunk)
            if decode_str(record[0]) == purchase_id:
                found = True
                continue
            tmp.write(chunk)
    if found:
        os.replace(temp_path, path)
        print("Purchase record deleted.")
    else:
        os.remove(temp_path)
        print("Purchase ID not found.")
```

รูปภาพที่ 4- 45 การลบข้อมูลการสั่งซื้อรายการเดียว

```
elif opt == 2:
    confirm = input("Are you sure you want to delete all purchases? (y/n): ")
    if confirm.lower() == "y":
        open(path, "wb").close()
        print("All purchase records deleted.")
    else:
        print("Cancelled.")
```

รูปภาพที่ 4- 46 การลบข้อมูลการสั่งซื้อทั้งหมด

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

ระบบจัดการคลังสินค้าที่พัฒนาขึ้นสามารถช่วยบริหารข้อมูลสินค้า การขาย และการสั่งซื้อได้อย่างมีประสิทธิภาพ โดยใช้การจัดเก็บข้อมูลแบบไฟล์ไบนารี พร้อมเมนูสำหรับ เพิ่ม แก้ไข ลบ และแสดงผลข้อมูลสินค้า ข้อมูลการขาย และการสั่งซื้อ ระบบยังสามารถตรวจสอบสถานะของสินค้า เช่น Active, Restock หรือ Deactive ตามจำนวนคงเหลือ และสามารถสร้างรายงานสรุป เช่น รายการสินค้าทั้งหมด ข้อมูลการขาย และข้อมูลการสั่งซื้อ ซึ่งช่วยให้การบริหารคลังสินค้าสะดวก รวดเร็ว และลดข้อผิดพลาดจากการบันทึกด้วยเอกสารกระดาษ

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

ในการพัฒนาระบบพบปัญหาหลักคือ การจัดการไฟล์ไบนารีที่ต้องใช้ โครงสร้างข้อมูลคงที่ (struct) ทำให้หากการเข้ารหัสหรือถอดรหัสไม่ถูกต้อง อาจเกิดข้อผิดพลาดได้ นอกจากนี้ยังพบข้อจำกัดในการแสดงผล เช่น ความยาวของชื่อสินค้า หมวดหมู่ หรือหน่วยที่ต้องถูกจำกัดตามขนาดที่กำหนด อีกทั้งระบบยังไม่มีการเชื่อมต่อฐานข้อมูลจริง ทำให้การจัดการข้อมูลจำนวนมากหรือการเข้าถึงพร้อมกันจากหลายผู้ใช้งานยังไม่สามารถทำได้เต็มที่

5.3 ข้อเสนอแนะ

เพื่อให้ระบบสมบูรณ์และพร้อมใช้งานจริง ควรปรับปรุงดังนี้

5.3.1 พัฒนาให้รองรับฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เช่น MySQL หรือ SQLite เพื่อจัดการข้อมูลจำนวนมากและรองรับผู้ใช้งานหลายคนพร้อมกัน

5.3.2 เพิ่มฟังก์ชันค้นหาและกรองข้อมูล เช่น ค้นหาสินค้าตามชื่อ หมวดหมู่ หรือรหัสสินค้า

5.3.3 ปรับปรุงระบบยืนยันตัวตนและจำกัดสิทธิ์การเข้าถึงของผู้ใช้แต่ละกลุ่ม

5.3.4 พัฒนาเป็นโปรแกรมที่มี ส่วนติดต่อผู้ใช้แบบกราฟิก (GUI) หรือเว็บแอปพลิเคชัน เพื่อความสะดวกและเข้าใจง่าย

5.4 สิ่งที่ผู้จัดทำได้รับในการพัฒนาโครงงาน

จากการพัฒนาระบบนี้ ผู้จัดทำได้รับความรู้และประสบการณ์ด้าน การออกแบบระบบ การเขียนโปรแกรมด้วย Python การจัดการไฟล์ไบนารี และการคิดวิเคราะห์เชิงตรรกะ นอกจากนี้ยังได้ฝึกทักษะการวางแผน การแบ่งงาน และการบริหารเวลา ทำให้มีความเข้าใจในกระบวนการพัฒนาระบบซอฟต์แวร์ และสามารถนำความรู้ไปประยุกต์ใช้ในโครงการหรืองานจริงในอนาคตได้