Project Term Assignment

โปรแกรม ระบบขายแล็ปท็อปมือสอง

รายวิชา Computer Programming รหัสรายวิชา 060233115

กลุ่ม We love Python

เสนอ

รศ.ดร อนิราช มิ่งขวัญ

สมาชิก

ปกป้อง ศกุนตนาค 6806022610348

นายภัทรกร มุขประดับ 6806022610178

เจษฎา กล้วยน้ำ 6806022610330

ชยณัฐ แข่งขัน 6806022610470

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ วิทยาเขตปราจีนบุรี

คณะเทคโนโลยีและการจัดการอุตสาหกรรม

ภาควิชาเทคโนโลยีสารสนเทศ สาขาวิชา วิศวกรรมสารสนเทศและเครือข่า

คำนำ

การจัดทำโครงงาน “ระบบขายแล็ปท็อปมือสอง” นี้เป็นส่วนหนึ่งของวิชา COMPUTER PROGRAMMIMG ของหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศคณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ เพื่อให้นักศึกษาได้นำความรู้ที่เรียนมาทั้งหมดมาประยุกต์ใช้ในการพัฒนาโปรแกรมที่สามารถทำงานได้จริง โดยเน้นการออกแบบและเขียนโปรแกรมในภาษา Python ซึ่งเป็นภาษาที่เรียนมาในวิชา COMPUTER PROGRAMMING โดยโครงงานนี้จะช่วย การคิดวิเคราะห์และแก้ปัญหาทางเทคนิค เพื่อเตรียมความพร้อมในการประกอบอาชีพด้านวิศวกรรมสารสนเทศและเครือข่ายในอนาคต หากมีข้อผิดพลาดประการใด คณะผู้จัดทำต้องขออภัยไว้ ณ ที่นี้ด้วย

บทที่ 1

บทนำ

1.1 วัตถุประสงค์ของโครงงาน

1.1.1 เพื่อพัฒนาระบบขายแล็ปท็อปมือสองอย่างมีประสิทธิภาพ

1.1.2 เพื่อฝึกฝนทักษะการเขียนโปรแกรมด้วย Python

1.1.3 เพื่อเรียนรู้วิธีการจัดการข้อมูลและไฟล์

1.1.4 เพื่อเรียนรู้การทำางานร่วมกันเป็นทีม

1.2 ขอบเขตของโครงงาน

1.2.1 ระบบขายแล็ปท็อปมือสองมีฟังก์ชั่นพื้นฐานทั้งหมด 15 ฟังก์ชั่น เช่น

1. เพิ่มลูกค้า
2. แก้ไขลูกค้า
3. ดูข้อมูลลูกค้า
4. ลบลูกค้า
5. กลับไปที่เมนู
6. เพิ่มโน้ตบุ๊ค
7. ลบโน้ตบุ๊ค
8. แก้ไขโน้ตบุ๊ค
9. ดูข้อมูลโน้ตบุ๊ค
10 เพิ่มรายการซื้อขาย
11. ลบรายการซื้อขาย
12. แก้ไขรายการซื้อขาย
13. ดูข้อมูลรายการซื้อขาย
14. สร้างรีพอร์ต(.txt)
15. เมนูออกจากหน้าปัจจุบัน

1.2.2 ระบบขายแล็ปท็อปมือสองประกอบด้วย 4 ไฟล์ได้แก่

1. cus_notebook.dat(ไฟล์ข้อมูลลูกค้า)
2. Info_notebook.dat(ไฟล์ข้อมูลโน้ตบุ๊ค)
3 sold_out.dat(ไฟล์ข้อมูลรายการขาย)
4. ไฟล์ report.txt

1.2.3 ระบบขายแล็ปท็อปมือสองมีการจัดเก็บข้อมูลหนังสือไว้ใน Text File ชื่อ report ซึ่งมีรหัสโน้ตบุ๊ค รหัสลูกค้า เบอร์โทร ที่อยู่ แบรนด์ ซีเรียล ปีวางจำหน่าย ราคา สถานะ ขาย จำนวนหนังสือทั้งหมด รายการผู้ยืมสถานะการยืม จำนวนหนังหนังสือที่ถูหยืม จำนวนหนังสือที่เหลือให้ยืม สถิติหนังสือในการยืม

1.2.4 ระบบขายแล็ปท็อปมือสองจะมีเมนูเพื่อให้ผู้ใช้สามารถเลือก ดำเนินการได้

1.3 ประโยชน์ที่ได้รับ

1.3.1 พัฒนาระบบขายแล็ปท็อปมือสองอย่างมีประสิทธิภาพ

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

ระบบขายแล็ปท็อปมือสอง

2.1 2 แฟ้มข้อมูลลูกค้าcus_notebook.dat

ไฟล์ข้อมูลลูกค้าประกอบด้วย 6 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด	ตัวอย่าง
1.customer_id	I	4	1001
2.name_cus	12s	12	Anirach
3.Addess_cus	24s	24	Prachinburi
4.brand	12s	12	Asus
5.model	16s	16	Rog-strix
6.tel	12s	12	08x-xxx-xxxx

ตารางที่ 2.1 แฟ้มข้อมูลหนังสือ

2.1.1 customer_id เลขที่สมาชิก

customer_id เป็นเลขที่สมาชิกที่ใช้ในการระบุสมาชิกแต่ละคนอย่างชัดเจนและไม่ซ้ำกันฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของตัวเลข (integer) เช่น 1001, 1002, 1003 เป็นต้น การมีเลขที่สมาชิกที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อหลีกเลี่ยงความสับสนระหว่างสมาชิกหลายคน และช่วยให้สามารถค้นหาและเรียกดูข้อมูลของสมาชิกได้อย่างแม่นยำและรวดเร็ว โดยเฉพาะในกรณีที่มีสมาชิกจำนวนมาก

2.1.2 name_cus ชื่อลูกค้า

name_cus คือ ชื่อลูกค้า ซึ่งฟิลด์นี้จะแสดงข้อมูลชื่อลูกค้าแต่ละคน ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (string) ตัวอย่างเช่น " ANIRACH " หรือ "POKPONG " การมีชื่อลูกค้าในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการเรียกดูข้อมูล ตรวจสอบการซื้อขายและทำการแก้ไขข้อมูลต่างๆ ลูกค้าแต่ละคนจะมีชื่อตามที่ระบุในการขาย และระบบจะใช้ชื่อดังกล่าวในการค้นหาและแสดงผลข้อมูลที่เกี่ยวข้องกับลูกค้าคนนั้น

2.1.3 addess_cus ที่อยู่ลูกค้า

addess_cus คือ ที่อยู่ลูกค้า ซึ่งฟิลด์นี้จะแสดงข้อมูลที่อยู่ลูกค้าแต่ละฟิลด์นี้เป็นประเภทข้อมูลข้อความ (string) “KMUTNB” หรือ “Prachinburi” การมีข้อมูลที่อยู่ลูกค้าในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการดูข้อมูลที่อยู่ลูกค้าแต่ละคน และทำการแก้ไขข้อมูลต่างๆ ลูกค้าแต่ละคนจะมีข้อมูลที่อยู่ลูกค้า หรือการถูกลบ และ ระบบจะใช้สถานะดังกล่าวในการแสดงผลข้อมูลที่เกี่ยวข้องกับลุกค้าคนนั้น

2.1.4 brand ยี่ห้อของแล็ปท็อป

brand คือ ยี่ห้อของแล็ปท็อปเครื่องนั้นๆ ซึ่งฟิลด์นี้จะแสดงข้อมูลยี่ห้อของแล็ปท็อปแต่ละเครื่อง ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (string) ตัวอย่างเช่น "LENOVO" หรือ "ASUS" การมียี่ห้อของแล็ปท็อปในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการเรียกดูข้อมูล ตรวจสอบแล็ปท็อป การซื้อขาย และทำการแก้ไขข้อมูลต่างๆ แลปท็อปแต่ละเครื่องจะมีชื่อตามที่ระบุในข้อมูล และ ระบบจะใช้ชื่อดังกล่าวในการค้นหาและแสดงผลข้อมูลที่เกี่ยวข้องกับแล็ปท็อป

2.1.5 .model รุ่นของแล็ปท็อป

medel คือ รุ่นของแล็ปท็อปที่ใช้ในการระบุรุ่นของแล็ปท็อปอย่างชัดเจน ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (string) เช่น “ROG”,”IDEAPAD”เป็นต้น การมีรุ่นของแล็ปท็อปที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อแสดงรุ่นของแล็ปท็อปเครื่องนั้นๆ และช่วยให้สามารถทราบรุ่นของแล็ปท็อปได้เพื่อเพิ่มความน่าเชื่อถือได้มากยิ่งขึ้น

2.1.6 tel หมายเลขโทรศัพท์ของลูกค้า

tel คือ ใช้เก็บหมายเลขโทรศัพท์ของลูกค้า ฟิลด์นี้เก็บในรูปแบบตัวเลขจำนวนเต็ม (integer) เช่น 0829481239 การเก็บหมายเลขโทรศัพท์ของลูกค้าจะช่วยในการบริหารจัดการทรัพยากร เช่น หมายเลขโทรศัพท์ของลูกค้าในการซื้อขาย

2.2 แฟ้มข้อมูลแล็ปท็อป Info_notebook.dat()

แฟ้มข้อมูลสมาชิกประกอบด้วย 6 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด	ตัวอย่าง
Notebook_id	I	4	1001
brand	12s	12	“Asus”
Serial_num	16s	16	“ABCD-1234”
rel	I	4	2024
price	F	4	200
Status (1=stock,0=sold out)	I	4	1/0

ตารางที่ 2.2 แฟ้มข้อมูลแล็ปท็อป

2.2.1 Notebook_id รหัสแล็ปท็อป

Notebook_id เป็นรหัสแล็ปท็อปที่ใช้ในการระบุแล็ปท็อปแต่ละเครื่องอย่างชัดเจนและไม่ซ้ำกันฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของตัวเลข (integer) เช่น 1001, 1002, 1003 เป็นต้น การมีรหัสแล็ปท็อปที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อหลีกเลี่ยงความสับสนระหว่างแล็ปท็อปหลายเครื่อง และช่วยให้สามารถค้นหาและเรียกดูข้อมูลของแล็ปท็อปได้อย่างแม่นยำและรวดเร็วโดยเฉพาะในกรณีที่มีแล็ปท็อปจำนวนมาก

2.2.2 brand ยี่ห้อของแล็ปท็อป

brand คือ ยี่ห้อของแล็ปท็อปเครื่องนั้นๆ ซึ่งฟิลด์นี้จะแสดงข้อมูลยี่ห้อของแล็ปท็อปแต่ละเครื่อง ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (string) ตัวอย่างเช่น "LENOVO" หรือ "ASUS" การมียี่ห้อของแล็ปท็อปในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการเรียกดูข้อมูล ตรวจสอบแล็ปท็อป การซื้อขาย และทำการแก้ไขข้อมูลต่างๆ แลปท็อปแต่ละเครื่องจะมีชื่อตามที่ระบุในข้อมูล และ ระบบจะใช้ชื่อดังกล่าวในการค้นหาและแสดงผลข้อมูลที่เกี่ยวข้องกับแล็ปท็อป

2.2.3 Serial_num เลขบัตรประชาชนของสินค้า

Serial_num คือ เลขบัตรประชาชนของสินค้า ซึ่งฟิลด์นี้จะแสดงข้อมูลเลขบัตรประชาชนของสินค้า ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (string) “ABCD-1234” หรือ “BACD-1234” การมีข้อมูลเลขบัตรประชาชนของสินค้าในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการดูข้อมูลเลขบัตรประชาชนของสินค้าแต่ละชิ้น และทำการแก้ไขข้อมูลต่างๆ หรือการถูกลบ และ ระบบจะใช้สถานะดังกล่าวในการแสดงผลข้อมูลที่เกี่ยวข้องกับสิ้นค้าชิ้นนั้น

2.2.4 rel ปีวางจำหน่าย

rel คือ ปีวางจำหน่าย ซึ่งฟิลด์นี้จะแสดงข้อมูลปีวางจำหน่ายแต่ละปี ฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของตัวเลข (integer) เช่น2021 , 2022, 2023 เป็นต้น การมีปีวางจำหน่ายในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการเรียกดูข้อมูล และทำการแก้ไขข้อมูลต่างๆ แลปท็อปแต่ละปีจะมีระบบจะใช้ปีดังกล่าวในการค้นหาและแสดงผลข้อมูลที่เกี่ยวข้องกับแลปท็อปเครื่องนั้น

2.2.5 . price ราคา

price คือ ราคา ฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของทศนิยม(float number) เช่น 20000.00 เป็นต้น การมีราคาที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อแสดงราคาในเครื่องนั้นๆ และช่วยให้สามารถทราบราคาได้เพื่อเพิ่มความน่าเชื่อถือได้มากยิ่งขึ้น

2.2.6 status สถานะ

status คือ สถานะแลปท็อปเครื่องนั้นๆ ซึ่งฟิลด์นี้จะแสดงข้อมูลสถานะของแล็ปท็อปแต่ละเครื่อง ฟิลด์นี้เป็นประเภทข้อมูลตัวเลขจำนวนเต็ม (integer) เช่น 1 = Active (ใช้งานอยู่/ยังมีในระบบ) หรือ0 = Deleted (ถูกลบ/ไม่ใช้งานแล้ว) การมีสถานะของหนังสือในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการดูสถานะของแลปท็อปแต่ละเครื่อง และทำการแก้ไขข้อมูลต่างๆ แล็ปท็อปแต่ละเครื่องจะมีสถานะตามที่ระบุในการใช้งานอยู่ หรือการถูกลบ และ ระบบจะใช้สถานะดังกล่าวในการแสดงผลข้อมูลที่เกี่ยวข้องกับแล็ปท็อปเครื่องนั้น

2.3 แฟ้มข้อมูลการขายออก sold_out.dat()

แฟ้มข้อมูลสมาชิกประกอบด้วย 6 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด	ตัวอย่าง
Sold_out_id	I	4	“1001”
Notebook_id	I	4	“1011”
Customer_id	I	4	“2001”
Soldout_date	12s	12	2025-10-01
Status (1=instock,0=soldout)	I	4	1/0

ตารางที่ 2.3 แฟ้มข้อมูลการขายออก

2.3.1 Sold_out_id รหัสซื้อขาย

Sold_out_id เป็นรหัสซื้อขายที่ใช้ในการระบุโน้ตบุ๊คแต่ละเครื่องอย่างชัดเจนและไม่ซ้ำกัน ฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของตัวเลข (integer) เช่น 1001, 1002, 1003 เป็นต้น การมีรหัสซื้อขายที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อหลีกเลี่ยงความสับสนระหว่างแลปท็อปหลายเครื่อง และช่วยให้สามารถค้นหาและเรียกดูข้อมูลของแลปท็อปได้อย่างแม่นยำและรวดเร็ว โดยเฉพาะในกรณีที่มีแลปท็อปจำนวนมาก

2.3.2 Notebook_id รหัสแล็ปท็อป

Notebook_id เป็นรหัสแล็ปท็อปที่ใช้ในการระบุแล็ปท็อปแต่ละเครื่องอย่างชัดเจและไม่ซ้ำกันฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของตัวเลข (integer) เช่น 1001, 1002, 1003 เป็นต้น การมีรหัสแล็ปท็อปที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อหลีกเลี่ยงความสับสนระหว่างแล็ปท็อปหลายเครื่อง และช่วยให้สามารถค้นหาและเรียกดูข้อมูลของแล็ปท็อปได้อย่างแม่นยำและรวดเร็ว โดยเฉพาะในกรณีที่มีแล็ปท็อปจำนวนมาก

2.3.3 customer_id เลขที่สมาชิก

customer_id เป็นเลขที่สมาชิกที่ใช้ในการระบุสมาชิกแต่ละคนอย่างชัดเจนและไม่ซ้ำกันฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของตัวเลข (integer) เช่น 1001, 1002, 1003 เป็นต้น การมีเลขที่สมาชิกที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อหลีกเลี่ยงความสับสนระหว่างสมาชิกหลายคน และช่วยให้สามารถค้นหาและเรียกดูข้อมูลของสมาชิกได้อย่างแม่นยำและรวดเร็ว โดยเฉพาะในกรณีที่มีสมาชิกจำนวนมาก

2.3.4 Soldout_date DD/MM/YY วันที่ซื้อขาย

Soldout_date DD/MM/YY วันที่ซื้อขาย ซึ่งฟิลด์นี้จะแสดงข้อมูลวันที่ซื้อขายแต่ละวัน ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (string) ตัวอย่างเช่น "12-01-1996 " หรือ "19-10-2012 " การมีวันที่ซื้อขายในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการเรียกดูข้อมูลตรวจสอบโน้ตบุ๊ค การซื้อขาย และทำการแก้ไขข้อมูลต่างๆ วันแต่ละวันระบบจะใช้วันดังกล่าวในการค้นหาและแสดงผลข้อมูลที่เกี่ยวข้องกับโน้ตบุ๊คเครื่องนั้น

2.3.5 . status สถานะ

status คือ สถานะแลปท็อปเครื่องนั้นๆ ซึ่งฟิลด์นี้จะแสดงข้อมูลสถานะของแล็ปท็อปแต่ละเครื่อง ฟิลด์นี้เป็นประเภทข้อมูลตัวเลขจำนวนเต็ม (integer) เช่น 1 = Active (ใช้งานอยู่/ยังมีในระบบ) หรือ0 = Deleted (ถูกลบ/ไม่ใช้งานแล้ว) การมีสถานะของหนังสือในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการดูสถานะของแลปท็อปแต่ละเครื่อง และทำการแก้ไขข้อมูลต่างๆ แล็ปท็อปแต่ละเครื่องจะมีสถานะตามที่ระบุในการใช้งานอยู่ หรือการถูกลบ และ ระบบจะใช้สถานะดังกล่าวในการแสดงผลข้อมูลที่เกี่ยวข้องกับแล็ปท็อปเครื่องนั้น

2.4 ไฟล์ report.txt

ไฟล์ report.txt ในระบบขายแล็ปท็อปมือสองของคุณประกอบด้วย 7 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

```
Notebook Store – Summary Report (Sample)
Generated At: 2025-10-03 10:15:54 (+07:00)
App Version: 1.0
Endianness: Little-Endian
Encoding: UTF-8 (fixed-length)

+------------+--------+------------+-----------------------+------------+----------------+------+------------+----------+------+
|NotebookID  |CusID   |Tel         |Address                |Brand       |Serial          |Year  |Price (THB) |Status    |Sold  |
+------------+--------+------------+-----------------------+------------+----------------+------+------------+----------+------+
|        1001|        |            |                       |Apple       |APLE-1234       |  2025|   900000.00|Active    |No    |
|        1002|    1002|0981237684  |LOPBURI                |Asus        |ASSP-1002       |  2024|    24000.00|Sold Out  |Yes   |
|        1003|        |            |                       |DELL        |DLLE-0913       |  2021|    20000.00|Active    |No    |
|        1004|        |            |                       |HP          |HPEE-1209       |  2011|    10000.00|Active    |No    |
|        1005|        |            |                       |Microsoft   |MCST-1291       |  2026|  1000000.00|Active    |No    |
|        1006|        |            |                       |MSI         |MSIQ-1239       |  2021|    45900.00|Active    |No    |
|        1007|        |            |                       |Asus        |ASSP-0943       |  2025|    90000.00|Active    |No    |
|        1008|    1001|0981238391  |KMUTNB Prachinburi     |Apple       |APLE-0133       |  2026| 10000000.00|Sold Out  |Yes   |
|        1009|        |            |                       |Acer        |ACEQ-0000       |  2019|    12299.00|Active    |No    |
|        1010|        |            |                       |Apple       |APLE-8239       |  2001|     1000.00|Active    |No    |
+------------+--------+------------+-----------------------+------------+----------------+------+------------+----------+------+

Summary (เฉพาะสถานะ Active)
- Total Notebooks (records): 10
- Active Notebooks: 10
- Deleted Notebooks: 0
- Currently Sold: 2
- Available Now: 8

Price Statistics (THB, Active only):
- Min : 1000.00
- Max : 10000000.00
- Avg : 1210319.90

Notebooks by Brand (Active only):
- Acer : 1
- Apple : 3
- Asus : 2
- DELL : 1
- HP : 1
- MSI : 1
- Microsoft : 1

Recent Activities:
[2025-10-03 10:15:52] Update Notebook id=1008

DEBUG: nb_to_cid = {1008: 1001, 1002: 1002}
DEBUG: cus_map keys = [1001, 1002, 1003]
```

รูปภาพที่ 2-1 ไฟล์ report

2.4.1 header_text ส่วนหัวรายงาน

ป็นฟิลด์ข้อความ (string 100 bytes) ใช้เก็บข้อความส่วนหัวของรายงาน เช่น "Library Borrow System – Summary Report" ฟิลด์นี้แสดงชื่อหรือวัตถุประสงค์ของรายงานเพื่อให้ผู้ใช้เข้าใจว่าเป็นรายงานประเภทใด

2.4.2 generated_at วันที่และเวลาที่สร้างรายงาน (YYYY-MM-DD HH:MM)

เป็นฟิลด์ข้อความ (string 25 bytes) ใช้เก็บวันและเวลาที่รายงานถูกสร้างขึ้นในรูปแบบ YYYY-MM-DD HH:MM เช่น "2025-10-01 09:30" ฟิลด์นี้ช่วยในการติดตามและอ้างอิงว่าไฟล์รายงานถูกสร้างเมื่อใด

2.4.3 app_version เวอร์ชันโปรแกรม เช่น "1.0"

เป็นฟิลด์ข้อความ (string 10 bytes) ใช้เก็บหมายเลขเวอร์ชันของโปรแกรมที่สร้างรายงาน เช่น "1.0", "2.1.5"ฟิลด์นี้มีประโยชน์ในการตรวจสอบว่าไฟล์รายงานถูกสร้างด้วยเวอร์ชั่นของระบบใด

2.4.4 encoding การเข้ารหัสไฟล์

เป็นฟิลด์ข้อความ (string 20 bytes) ใช้ระบุรูปแบบการเข้ารหัสไฟล์ เช่น "UTF-8", "ISO-8859-1" เพื่อให้การอ่านไฟล์รายงานถูกต้องตรงกับภาษาที่ใช้งาน

2.4.5 notebook_table_header หัวตาราง notebook_table_header

เป็นฟิลด์ข้อความ (string 80 bytes) ใช้เก็บหัวตารางสำหรับแสดงข้อมูลหนังสือ เช่น " NotebookID |CusID |Tel |Address |Brand |Serial |Year |Price (THB) |Status|Sold" เพื่อกำหนดโครงสร้างของตารางในส่วนรายงาน

2.4.6 notebook_records ข้อมูลตาราง

เป็นฟิลด์ข้อความ (string 120 * N bytes, โดย N = จำนวนแล็ปท็อป) ใช้เก็บข้อมูลแลปท็อปแต่ละเครื่องในรูปแบบความยาวคงที่ (fixed-length record) เช่น รายการรหัสแลปท็อป, รหัสสมาชิก,เบอร์โทร,ที่อยู่,แบรนด์ ซีเรียล ปี ราคา สถานะ และสถานะการขาย

2.4.7 summary_section สรุปข้อมูล

เป็นฟิลด์ข้อความ (string 150 bytes) ใช้เก็บข้อมูลสรุปของระบบ เช่น Total = จำนวนแลปท็อปทั้งหมด Active = รายการแลปท็อปที่ใช้งานอยู่ Deleted = แลปท็อปที่ถูกลบ Currently Sold = แลปท้อปที่ขายออกไปล่าสุด Available = แลปท็อปที่ว่าง ช่วยให้ผู้ใช้เห็นภาพรวมของฐานข้อมูลหนังสือได้อย่างรวดเร็ว

2.4.8 Price Statistics (THB, Active only) สถิติราคาขายเป็นฟิลด์ข้อความ (string 150 bytes) ใช้เก็บข้อมูลสถิติการยืม เช่น min= ราคาขายต่ำสุด max = ราคาขายที่สูงที่สุด avg= ราคาค่าเฉลี่ย ฟิลด์นี้มีประโยชน์ต่อการวิเคราะห์ราคาต่ำสุดและสูงสุด

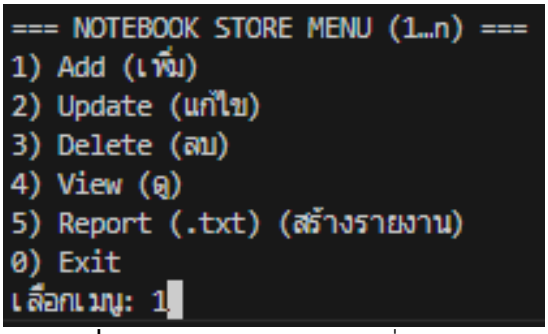
บทที่ 3
การใช้งานระบบขายแล็ปท็อปมือสอง

โปรแกรมระบบขายแล็ปท็อปมือสองคือการช่วยการขายแลปท็อปให้สะดวกและง่ายขึ้นและยังช่วยจัดประเภทของแลปท็อปให้ดูง่ายขึ้นโดยการช่วยทำรายงานไปเก็บไว้ยังไฟล์Text โปรแกรมขายแล็ปท็อปมือสองประกอบไปด้วย การเพิ่มข้อมูลที่จะเก็บ NotebookID ,CusID ,Tel ,Address ,Brand ,Serial ,Year ,Price (THB),Status,Sold แสดงข้อมูลแลปท็อปทั้งหมด ในโปรแกรมค้นหาข้อมูลโดยใช้ NotebookID และ CusID ในการค้นหาอัพเดทข้อมูลสามารถเปลี่ยนแปลงข้อมูลได้ แต่ถ้าไม่ต้องการแก้ไขข้อมูลบางส่วนสามารถกด Enter เพื่อข้ามการเปลี่ยนแปลงในข้อมูลนั้นๆ ได้เลยลบข้อมูลโดยการใช้เลข NotebookID เพื่อลบข้อมูลทั้งหมดของเลข NotebookID นั้น สร้างรายงานเพื่อทำสรุปการการขายแลปท็อปที่มีข้อมูลทั้งหมดในโปรแกรมจบการทำงานของโปรแกรม

สำหรับผู้ใช้งานโปรแกรม

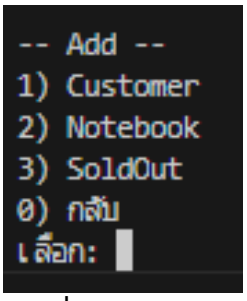
3.1 การใช้งานโปรแกรมระบบขายแล็ปท็อปมือสอง

3.1.1 กรอกหมายเลข 1 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Add เพิ่มข้อมูลที่ประกอบไปด้วย Add Customer, Add note , Add sold out ,Back to Main Menu



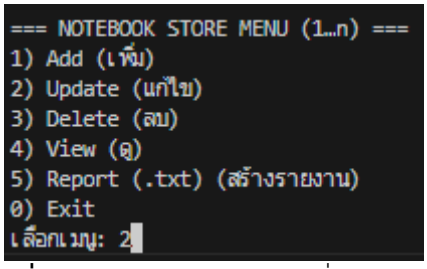
รูปภาพที่ 3-1 การเลือกใช้งานฟังก์ชั่น notebook

3.1.2 เมื่อเมนูฟังก์ชั่น Manage Book ขึ้นมาแล้วจากนั้นก็สามารถระบุเมนูที่ต้องการเลือกได้



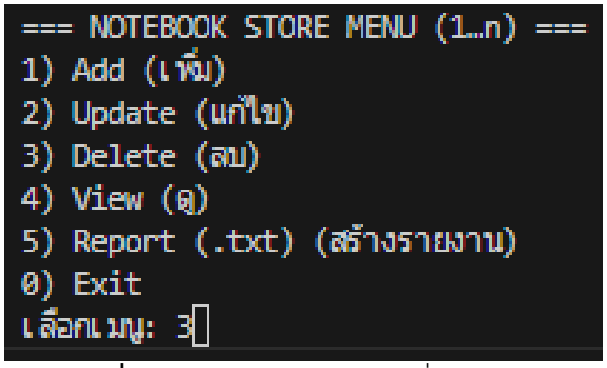
รูปภาพที่ 3-2 เมนูของ Add

3.1.3 กรอกหมายเลข 2 เพื่อเรียกฟังก์ชัน Update เพิ่มข้อมูลที่ประกอบไปด้วยupdate customer,update notebook, update sold out , Back to Main Menu



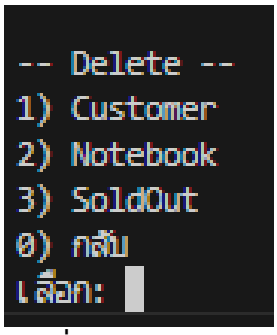
รูปภาพที่ 3-3 การเลือกใช้งานฟังก์ชั่นของ Update

3.1.5 กรอกหมายเลข 3 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน delete เพิ่มข้อมูลที่ประกอบไปด้วยdelete Customer,delete notebook, delete soldout, Back to MainMenu



รูปภาพที่ 3- 5 การเลือกใช้งานฟังก์ชั่นของ delete

3.1.6 เมื่อเมนูฟังก์ชั่น Manage Loans ขึ้นมาแล้วจากนั้นก็สามารถระบุเมนูที่ต้องการเลือกได้



รูปภาพที่ 3-6 เมนูของ delete

3.1.7 กรอกหมายเลข 4 เพื่อเรียกฟังก์ชัน view ดูข้อมูลที่ประกอบไปด้วย view,Customer,view notebook, view soldout, Back to Main Menu

```
=== NOTEBOOK STORE MENU (1…n) ===
1) Add (เพิ่ม)
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Report (.txt) (สร้างรายงาน)
0) Exit
เลือกเมนู: 4
```

รูปภาพที่ 3-7 การเลือกใช้งานฟังก์ชั่นของ view

3.1.8 กรอกหมายเลข 5 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Exit เพื่อออกจากโปรแกรม

```
-- View --
1) Customer
2) Notebook
3) SoldOut
0) กลับ
เลือก:
```

รูปภาพที่ 3- 8 เมนูของ view

3.1.9 กรอกหมายเลข 5 เพื่อเรียกฟังก์ชัน report(.txt) เพื่อเพิ่มไฟล์ report.txt ที่สามารถเขียน report ถึงห้องสมุดได้

```
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Report (.txt) (สร้างรายงาน)
0) Exit
เลือกเมนู: 5
[2025-10-06 10:30:27] Report written: report.txt
Report saved to report.txt
```

รูปภาพที่ 3- 9 การเลือกใช้งานฟังก์ชั่นของ report.txt

3.1.8 กรอกหมายเลข 6 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Exit เพื่อออกจากโปรแกรม

```
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Report (.txt) (สร้างรายงาน)
0) Exit
เลือกเมนู: 5
[2025-10-06 10:30:27] Report written: report.txt
Report saved to report.txt
```

รูปภาพที่ 3- 9 การเลือกใช้งานฟังก์ชั่นของ Exit

3.2 การใช้งานโปรแกรมเพิ่มข้อมูล

3.2.1 กรอกหมายเลข 1 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Add เพิ่มข้อมูลที่ประกอบไปด้วย Add Customer, Add note , Add sold out ,Back to Main Menu

```
=== NOTEBOOK STORE MENU (1…n) ===
1) Add (เพิ่ม)
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Report (.txt) (สร้างรายงาน)
0) Exit
เลือกเมนู: 1
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่น Book

3.2.2 เมื่อกดเลือกหมายเลข 1 จะปรากฏหัวข้อการใส่รหัสสมาชิกและจากนั้นใส่ข้อมูลในหัวข้อทั้งหมดดังภาพที่ 3-10

```
เลือก: 1
ระบุ customer_id (int): 1006
ชื่อ (<=12 bytes): pokpong
ที่อยู่ (<=24 bytes): prachinburi
แบรนด์ (<=12 bytes): lenovo
รุ่น (<=16 bytes): loq
โทร (<=12 bytes): 0897637819
[2025-10-06 10:56:09] Add Customer id=1006, name=pokpong
```

รูปภาพที่ 3- เมนูของ Add customer

3.2.3 กรอกหมายเลข 2 เพื่อเรียกฟังก์ชัน Add notebook ปรากฏหัวข้อการใส่รหัสสมาชิก และจากนั้นใส่ข้อมูล

```
-- Add --
1) Customer
2) Notebook
3) SoldOut
0) กลับ
เลือก: 2
ระบุ notebook_id (int): 1011
แบรนด์ (<=12 bytes): Lonovo
ซีเรียล (<=16 bytes): ASOQ-1943
ปี/รุ่น (rel:int): 2025
ราคา (float): 2000000
สถานะสินค้า 1=stock, 0=sold out (0/1): 1
[2025-10-06 10:59:47] Add Notebook id=1011, brand=Lonovo, status=1
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่นของ add notebook

3.2.4 กรอกหมายเลข 3 เมื่อเมนูฟังก์ชั่น Add soldout ปรากฏหัวข้อการใส่รหัสสมาชิกและจากนั้นใส่ข้อมูล

```
-- Add --
1) Customer
2) Notebook
3) SoldOut
0) กลับ
เลือก: 3
sold_out_id: 1010
notebook_id: 1011
customer_id: 1006
Name Customer: pokpong1234
Sold date: 2025-10-05
สถานะ 1=instock, 0=soldout (ตามสเปคไฟล์) (0/1): 0
** Warning: ไม่สามารถอัปเดตสถานะโน้ตบุ๊กได้: ไม่พบ ID 0
[2025-10-06 11:03:07] Add Soldout id=1010, nid=1011, cid=1006
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่นของ add soldout

3.3 การใช้งานโปรแกรมแก้ไขข้อมูล

3.3.1 กรอกหมายเลข 2 เพื่อเรียกฟังก์ชัน Update เพิ่มข้อมูลที่ประกอบไปด้วยupdate customer,update notebook, update sold out , Back to Main Menu

```
=== NOTEBOOK STORE MENU (1…n) ===
1) Add (เพิ่ม)
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Report (.txt) (สร้างรายงาน)
0) Exit
เลือกเมนู: 2
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่นของ Update

3.3.2 กรอกหมายเลข 1 เพื่อเรียกเมนูฟังก์ชั่น Update customer ปรากฏหัวข้อการใส่รหัสสมาชิก และจากนั้นใส่ข้อมูล

```
-- Update --
1) Customer
2) Notebook
3) SoldOut
0) กลับ
เลือก: 1
ระบุ customer_id ที่ต้องการแก้ไข: 1006
ข้อมูลเดิม: {'customer_id': 1006, 'name': 'pokpong', 'address': 'prachinburi', 'brand': 'lenovo', 'model': 'loq', 'tel': '0897637819'}
ชื่อ [pokpong]: tle
ที่อยู่ [prachinburi]: chonburi
แบรนด์ [lenovo]:
รุ่น [loq]:
โทร [0897637819]: 0728192019
[2025-10-06 11:06:13] Update Customer id=1006
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่นของ Update customer

3.3.3 กรอกหมายเลข 2 เพื่อเรียกเมนูฟังก์ชั่น Update customer ปรากฏหัวข้อการใส่รหัสสมาชิก และจากนั้นใส่ข้อมูล

```
-- Update --
1) Customer
2) Notebook
3) SoldOut
0) กลับ
เลือก: 2
ระบุ notebook_id ที่ต้องการแก้ไข: 1011
ข้อมูลเดิม: {'notebook_id': 1011, 'brand': 'Lonovo', 'serial_num': 'ASOQ-1943', 'rel': 2025, 'price': 2000000.0, 'status': 1}
แบรนด์ [Lonovo]:
ซีเรียล [ASOQ-1943]:
rel [2025]: 2024
ราคา [2000000.0]:
สถานะ (1=stock,0=sold) [1]:
[2025-10-06 11:10:18] Update Notebook id=1011
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่นของ Update notebook

3.3.3 กรอกหมายเลข 3 เพื่อเรียกเมนูฟังก์ชั่น Update soldout ปรากฏหัวข้อการใส่รหัสสมาชิกและจากนั้นใส่ข้อมูล

```
ระบุ sold_out_id ที่ต้องการแก้ไข: 1010
ข้อมูลเดิม: {'sold_out_id': 1010, 'notebook_id': 1011, 'customer_id': 1006, 'name': 'pokpong112', 'soldout_date': '2025-10-05', 'status': 0}
notebook_id [1011]:
** กรุณากรอกตัวเลข **
notebook_id [1011]:
** กรุณากรอกตัวเลข **
notebook_id [1011]: 1011
customer_id [1006]: 1006
ชื่อลูกค้า [pokpong112]: Pokpong
วันที่ขาย [2025-10-05]:
สถานะ (1/0) [0]:
** Warning: ไม่สามารถอัปเดตสถานะโน้ตบุ๊กได้: ไม่พบ ID 0
[2025-10-06 11:29:33] Update Soldout id=1010
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่นของ Update soldout

3.4 การใช้งานโปรแกรมแสดงข้อมูล

3.4.1 กรอกหมายเลข 4 เพื่อเรียกฟังก์ชัน view ดูข้อมูลที่ประกอบไปด้วยview Customer,view notebook, view soldout, Back to Main Menu

```
=== NOTEBOOK STORE MENU (1…n) ===
1) Add (เพิ่ม)
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Report (.txt) (สร้างรายงาน)
0) Exit
เลือกเมนู: 4
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่น view

3.2.2 กรอกหมายเลข2 เพื่อเรียกเมนูฟังก์ชั่น view all customer ปรากฏหัวข้อการใส่รหัสสมาชิกและจากนั้นใส่ข้อมูล

```
-- ดูข้อมูลลูกค้า --
1) ดูรายการเดียว (by id)
2) ดูทั้งหมด
3) ดูแบบกรอง (by brand)
4) สถิติโดยสรุป
0) กลับเมนูหลัก
เลือก: 2
{'customer_id': 1006, 'name': 'tle', 'address': 'chonburi', 'brand': 'lenovo', 'model': 'loq', 'tel': '0728192019'}
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่นของ view customer

3.2.3 กรอกหมายเลข2 เพื่อเรียกเมนูฟังก์ชั่น view all notebook ปรากฏหัวข้อการใส่รหัสสมาชิกและจากนั้นใส่ข้อมูล

```
-- ดูข้อมูลการขาย --
1) ดูรายการเดียว (by id)
2) ดูทั้งหมด
3) ดูแบบกรอง (by วันที่ขาย หรือสถานะ)
4) สถิติโดยสรุป
0) กลับเมนูหลัก
เลือก: 2
{'sold_out_id': 1010, 'notebook_id': 1011, 'customer_id': 1006, 'name': 'Pokpong', 'soldout_date': '2025-10-05', 'status': 0}
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่นของ view notebook

3.2.4 กรอกหมายเลข2 เพื่อเรียกเมนูฟังก์ชั่น view all notebook ปรากฏหัวข้อการใส่รหัสสมาชิกและจากนั้นใส่ข้อมูล

```
-- ดูข้อมูลการขาย --
1) ดูรายการเดียว (by id)
2) ดูทั้งหมด
3) ดูแบบกรอง (by วันที่ขาย หรือสถานะ)
4) สถิติโดยสรุป
0) กลับเมนูหลัก
เลือก: 2
{'sold_out_id': 1010, 'notebook_id': 1011, 'customer_id': 1006, 'name': 'Pokpong', 'soldout_date': '2025-10-05', 'status': 0}
```

รูปภาพที่ 3- การเลือกใช้งานฟังก์ชั่นของ view soldout

บทที่ 4
อธิบายการทำงานของ Code

4.1 ฟังก์ชั่นไบนารีพื้นฐานในระบบซื้อขายแลปท็อป

4.1.1 Module Struct เป็นโมดูลใน Python ที่ใช้สำหรับการจัดการข้อมูลแบบไบนารีเช่น การแปลงข้อมูลจากรูปแบบ Python (เช่น integer, float) ไปเป็นไบต์ หรือการแปลงข้อมูลจากไบต์กลับมาเป็นรูปแบบ Python อีกครั้ง โมดูลนี้สำคัญเมื่อเราต้องการทำงานกับไฟล์หรือข้อมูลที่อยู่ในรูปแบบไบนารี เช่นไฟล์

```
import struct
```

รูปภาพที่ 4-1 Code Module Pickle

4.1.2 import os เป็นโมดูลมาตรฐานของ Python ที่ใช้สำหรับจัดการกับระบบปฏิบัติการ เช่น การเข้าถึงไฟล์และโฟลเดอร์ การจัดการ path และการลบไฟล์

```
import os
```

รูปภาพที่ 4-2 Code Module os

4.1.3 from datetime import ใช้สำหรับจัดการกับวันและเวลา โดยสามารถดึงเวลาปัจจุบัน แปลงข้อความเป็นวันที่ หรือแปลงวันที่เป็นข้อความได้

```
from datetime import datetime
```

รูปภาพที่ 4-3 Code Module Time

4.1.4 timezone ใช้สร้าง เขตเวลา (time zone) แบบ offset จาก UTC

4.1.5 timedelta ใช้แทน ช่วงเวลา (days, hours, minutes, seconds ฯลฯ) เราสามารถใช้บวก/ลบกับ datetime ได้

4.1.6 โครงสร้างข้อมูล notebook struck "<i50si30siiil" คือ format string กำหนด layout ของrecord

4.1.7 ฟังก์ชัน _ensure_file(self)

ฟังก์ชันนี้ใช้สำหรับตรวจสอบว่าไฟล์ที่กำหนดไว้ใน self.path มีอยู่แล้วหรือไม่ หากยังไม่มีไฟล์ดังกล่าว จะทำการสร้างไฟล์ใหม่ขึ้นมาในโหมดเขียนแบบไบต์ ('wb') โดยไม่เขียนข้อมูลใด ๆ ลงไปในไฟล์นั้น

```
def _ensure_file(self):
    if not os.path.exists(self.path):
        with open(self.path, 'wb') as f:
            pass
```

รูปภาพที่ 4-2 ฟังก์ชั่น add_book

4.2 ฟังก์ชั่นเมนูระบบซื้อขายแลปท็อปมือสอง

4.2.1 ฟังก์ชัน add_customer

ฟังก์ชัน ฟังก์ชัน add_customer ฟังก์ชันนี้ใช้สำหรับเพิ่มข้อมูลลูกค้าใหม่เข้าสู่ฐานข้อมูล โดยรับข้อมูลจากผู้ใช้ผ่านทางคีย์บอร์ด แล้วจัดรูปแบบข้อมูลให้อยู่ในรูปแบบไบต์ (binary) ก่อนบันทึกใช้ฟังก์ชัน input_int() เพื่อรับค่าตัวเลขจำนวนเต็มจากผู้ใช้กำหนดว่าไม่อนุญาตให้เป็นศูนย์ (allow_zero=False) และต้องเป็นค่าบวกเท่านั้น (positive_only=True)ใช้ฟังก์ชัน input_fixed_str() เพื่อรับข้อความจากผู้ใช้ โดยจำกัดขนาดเป็นจำนวนไบต์ที่กำหนดช่วยให้ข้อมูลมีขนาดคงที่เมื่อจัดเก็บในไฟล์หรือฐานข้อมูลใช้ฟังก์ชัน pack_customer() เพื่อรวมข้อมูลทั้งหมดให้อยู่ในรูปแบบไบต์

```
def add_customer():
    cid = input_int("ระบุ customer_id (int): ", allow_zero=False, positive_only=True)
    name = input_fixed_str("ชื่อ (<=12 bytes): ", 12)
    addr = input_fixed_str("ที่อยู่ (<=24 bytes): ", 24)
    brand = input_fixed_str("แบรนด์ (<=12 bytes): ", 12)
    model = input_fixed_str("รุ่น (<=16 bytes): ", 16)
    tel = input_fixed_str("โทร (<=12 bytes): ", 12)
    packed = pack_customer(0, cid, name, addr, brand, model, tel)
    cus_db.add(packed, cid)
    log_action(f"Add Customer id={cid}, name={name}")
```

รูปภาพที่ 4- ฟังก์ชั่น menu_add_customer

4.2.1 ฟังก์ชัน update_customer

ใช้สำหรับแก้ไขข้อมูลลูกค้า โดยรับ customer_id จากผู้ใช้ แล้วดึงข้อมูลเดิมมาแสดง จากนั้นให้ผู้ใช้กรอกข้อมูลใหม่ หากไม่กรอกจะใช้ข้อมูลเดิมแทน แล้วบันทึกข้อมูลใหม่กลับไปยังฐานข้อมูลใช้ input_int() รับรหัสลูกค้าตรวจสอบว่ามีข้อมูลหรือไม่ใช้ unpack_customer() เพื่อแปลงข้อมูลจากไบต์ใช้ input_fixed_str() รับข้อมูลใหม่ โดยมีค่าเริ่มต้นเป็นข้อมูลเดิมใช้ pack_customer() เพื่อจัดรูปแบบข้อมูลใหม่บันทึกข้อมูลด้วย cus_db.update()บันทึกการกระทำด้วย log_action()

```
def update_customer():
    cid = input_int("ระบุ customer_id ที่ต้องการแก้ไข: ", allow_zero=False, positive_only=True)
    offset, rec = cus_db.get(cid)
    if rec is None:
        print("** ไม่พบข้อมูล **")
        return
    data = unpack_customer(rec)
    print("ข้อมูลเดิม:", data)
    name = input_fixed_str(f"ชื่อ [{data['name']}]: ", 12) or data['name']
    addr = input_fixed_str(f"ที่อยู่ [{data['address']}]: ", 24) or data['address']
    brand = input_fixed_str(f"แบรนด์ [{data['brand']}]: ", 12) or data['brand']
    model = input_fixed_str(f"รุ่น [{data['model']}]: ", 16) or data['model']
    tel = input_fixed_str(f"โทร [{data['tel']}]: ", 12) or data['tel']
    packed = pack_customer(0, cid, name, addr, brand, model, tel)
    cus_db.update(cid, packed)
    log_action(f"Update Customer id={cid}")
```

รูปภาพที่ 4- ฟังก์ชั่น menu_update_customer

4.2.2 ฟังก์ชัน delete_customer

ใช้สำหรับลบข้อมูลลูกค้า โดยรับ customer_id จากผู้ใช้ แล้วเรียกใช้เมธอด delete() เพื่อลบข้อมูลจากฐานข้อมูลใช้ input_int() รับรหัสลูกค้าลบข้อมูลด้วย cus_db.delete()บันทึกการกระทำด้วยlog_action()

```
def delete_customer():
    cid = input_int("ระบุ customer_id ที่ต้องการลบ: ", allow_zero=False, positive_only=True)
    cus_db.delete(cid)
    log_action(f"Delete Customer id={cid}")
```

รูปภาพที่ 4- ฟังก์ชั่น menu_delete_customer

4.2.3 ฟังก์ชัน view_customer_menu

ใช้สำหรับแสดงเมนูการดูข้อมูลลูกค้า โดยให้ผู้ใช้เลือกว่าจะดูข้อมูลตามรหัสลูกค้า หรือดูตามแบรนด์ เป็นต้น

```python
def view_customer_menu():
    print("\n-- ดูข้อมูลลูกค้า --")
    print("1) ดูรายการเดียว (by id)")
    print("2) ดูทั้งหมด")
    print("3) ดูแบบกรอง (by brand)")
    print("4) สถิติโดยสรุป")
    print("0) กลับเมนูหลัก")
    choice = input("เลือก: ").strip()
    if choice == '1':
        cid = input_int("ระบุ customer_id: ", allow_zero=False, positive_only=True)
        offset, rec = cus_db.get(cid)
        if rec is None:
            print("** ไม่พบข้อมูล **")
            return
        print(unpack_customer(rec))
    elif choice == '2':
        for _, rec in cus_db.iter_active():
            print(unpack_customer(rec))
    elif choice == '3':
        brand = input("ระบุแบรนด์ที่ต้องการกรอง: ").strip()
        for _, rec in cus_db.iter_active():
            d = unpack_customer(rec)
            if d['brand'].lower() == brand.lower():
                print(d)
    elif choice == '4':
        s = cus_db.stats()
        print("สรุปลูกค้า:", s)
```

รูปภาพที่ 4- ฟังก์ชั่น menu_view_customer

4.2.4 ฟังก์ชัน add_notebook

ใช้สำหรับเพิ่มข้อมูลโน้ตบุ๊กใหม่เข้าสู่ฐานข้อมูล โดยรับข้อมูลจากผู้ใช้ เช่น รหัสโน้ตบุ๊ก, ยี่ห้อ, หมายเลขเครื่อง, หมายเลขอ้างอิง, ราคา และสถานะ รับข้อมูลผ่านฟังก์ชัน input_int() และ input_fixed_str()ใช้ pack_notebook() เพื่อจัดรูปแบบข้อมูลให้อยู่ในรูปแบบไบต์เพิ่มข้อมูลลงฐาน

ข้อมูลด้วย nb_db.add()บันทึกการกระทำด้วย log_action()

```python
def add_notebook():
    nid = input_int("ระบุ notebook_id (int): ", allow_zero=False, positive_only=True)
    brand = input_fixed_str("แบรนด์ (<=12 bytes): ", 12)
    serial = input_fixed_str("ซีเรียล (<=16 bytes): ", 16)
    rel = input_int("ปี/รุ่น (rel:int): ", allow_zero=True, positive_only=False)
    price = input_float("ราคา (float): ", positive_only=True)
    status = input_status("สถานะสินค้า 1=stock, 0=sold out")
    packed = pack_notebook(0, nid, brand, serial, rel, price, status)
    nb_db.add(packed, nid)
    log_action(f"Add Notebook id={nid}, brand={brand}, status={status}")
```

รูปภาพที่ 4- ฟังก์ชั่น menu_ add_notebook

4.2.5ฟังก์ชัน update_notebook

ใช้สำหรับแก้ไขข้อมูลโน้ตบุ๊กที่มีอยู่ โดยรับรหัสโน้ตบุ๊กจากผู้ใช้ แล้วดึงข้อมูลเดิมมาแสดง จากนั้นให้ผู้ใช้กรอกข้อมูลใหม่ หากไม่กรอกจะใช้ข้อมูลเดิมแทนแล้วบันทึกข้อมูลใหม่กลับไปยังฐานข้อมูลตรวจสอบว่าข้อมูลมีอยู่หรือไม่ใช้ unpack_notebook() เพื่อแปลงข้อมูลจากไบต์รับข้อมูลใหม่จากผู้ใช้ พร้อมแสดงค่าปัจจุบันเป็นค่าเริ่มต้นใช้ pack_notebook() เพื่อจัดรูปแบบข้อมูลใหม่บันทึกข้อมูลด้วย nb_db.update()บันทึกการกระทำด้วย log_action()

```
def update_notebook():
    nid = input_int("ระบุ notebook_id ที่ต้องการแก้ไข: ", allow_zero=False, positive_only=True)
    offset, rec = nb_db.get(nid)
    if rec is None:
        print("** ไม่พบข้อมูล **")
        return
    data = unpack_notebook(rec)
    print("ข้อมูลเดิม:", data)
    brand = input_fixed_str(f"แบรนด์ [{data['brand']}]: ", 12) or data['brand']
    serial = input_fixed_str(f"ซีเรียล [{data['serial_num']}]: ", 16) or data['serial_num']
    rel = input_int(f"rel [{data['rel']}]: ", allow_zero=True, positive_only=False)
    price_in = input(f"ราคา [{data['price']}]: ").strip()
    price = float(price_in) if price_in else data['price']
    status_in = input(f"สถานะ (1=stock,0=sold) [{data['status']}]: ").strip()
    status = int(status_in) if status_in in ('0', '1') else data['status']
    packed = pack_notebook(0, nid, brand, serial, rel, price, status)
    nb_db.update(nid, packed)
    log_action(f"Update Notebook id={nid}")
```

รูปภาพที่ 4- ฟังก์ชั่น menu_ update_notebook

4.2.6 ฟังก์ชัน delete_notebook

ใช้สำหรับลบข้อมูลโน้ตบุ๊ก โดยรับรหัสโน้ตบุ๊กจากผู้ใช้ แล้วเรียกใช้เมธอด delete() เพื่อลบข้อมูลจากฐานข้อมูล

```
def delete_notebook():
    nid = input_int("ระบุ notebook_id ที่ต้องการลบ: ", allow_zero=False, positive_only=True)
    nb_db.delete(nid)
    log_action(f"Delete Notebook id={nid}")
```

รูปภาพที่ 4- ฟังก์ชั่น menu_ delete_notebook

4.2.7 ฟังก์ชัน view_notebook_menu

ใช้สำหรับแสดงเมนูการดูข้อมูลโน้ตบุ๊ก โดยให้ผู้ใช้เลือกว่าจะดูข้อมูลทั้งหมด หรือกรองตามเงื่อนไข เช่น ยี่ห้อ, สถานะ, ช่วงราคา หรือดูสถิติใช้ print() แสดงเมนูให้ผู้ใช้เลือกใช้ nb_db.iter_active() เพื่อวนลูปข้อมูลโน้ตบุ๊กที่ยังใช้งานอยู่มีตัวเลือกการกรองข้อมูลเช่น:ดูทั้งหมด ดูตามยี่ห้อดูตามสถานะ (ขายแล้วยังอยู่ในสต็อก)ดูตามช่วงราคาดูสถิติสินค้าคงคลังและสินค้าที่ขายไป

```
def view_notebook_menu():
        for _, rec in nb_db.iter_active():
            print(unpack_notebook(rec))
    elif choice == '3':
        print("กรอง: 1=brand, 2=status, 3=ช่วงราคา")
        g = input("เลือกตัวกรอง: ").strip()
        if g == '1':
            brand = input("ระบุแบรนด์: ").strip()
            for _, rec in nb_db.iter_active():
                d = unpack_notebook(rec)
                if d['brand'].lower() == brand.lower():
                    print(d)
        elif g == '2':
            st = input_status("สถานะที่ต้องการ (1=stock,0=sold)")
            for _, rec in nb_db.iter_active():
                d = unpack_notebook(rec)
                if d['status'] == st:
                    print(d)
        elif g == '3':
            pmin = input_float("ราคา MIN: ", positive_only=True)
            pmax = input_float("ราคา MAX: ", positive_only=True)
            if pmin > pmax:
                pmin, pmax = pmax, pmin
            for _, rec in nb_db.iter_active():
                d = unpack_notebook(rec)
                if pmin <= d['price'] <= pmax:
                    print(d)
    elif choice == '4':
        s = nb_db.stats()
        stock = 0
        sold = 0
        for _, rec in nb_db.iter_active():
            d = unpack_notebook(rec)
            if d['status'] == 1:
                stock += 1
            else:
                sold += 1
        print("สรุปโน้ตบุ๊ก:", s, "| stock=", stock, "sold_out=", sold)
```

รูปภาพที่ 4- ฟังก์ชั่น menu_ view_notebook

4.2.8 ฟังก์ชัน add_soldout

ใช้สำหรับเพิ่มข้อมูลการขายโน้ตบุ๊กลงในฐานข้อมูล โดยรับข้อมูลจากผู้ใช้ เช่น รหัสการขาย, รหัสโน้ตบุ๊ก, รหัสลูกค้า, ชื่อลูกค้า, วันที่ขาย และสถานะ จากนั้นบันทึกข้อมูลและอัปเดตสถานะของโน้ตบุ๊กที่เกี่ยวข้องใช้ input_int() และ input_fixed_str() รับข้อมูลจากผู้ใช้ใช้ pack_soldout() เพื่อจัดรูปแบบข้อมูลให้อยู่ในรูปแบบไบต์เพิ่มข้อมูลลงฐานข้อมูลด้วย so_db.add()ตรวจสอบและอัปเดตสถานะของโน้ตบุ๊กที่ถูกขายผ่าน nb_db.update()บันทึกการกระทำด้วย log_action()

```python
def add_soldout():
    sid = input_int("sold_out_id: ", allow_zero=False, positive_only=True)
    nid = input_int("notebook_id: ", allow_zero=False, positive_only=True)
    cid = input_int("customer_id: ", allow_zero=False, positive_only=True)
    name = input_fixed_str("Name Customer: ", 12)
    sold_date = input_fixed_str("Sold date: ", 12)
    status = input_status("สถานะ 1=instock, 0=soldout (ตามสเปคไฟล์)")
    packed = pack_soldout(0, sid, nid, cid, name, sold_date, status)
    so_db.add(packed, sid)

    # อัปเดตสถานะโน้ตบุ๊กตามการขาย — ใช้ offset ที่ได้จาก nb_db.get()
    try:
        off, nbrec = nb_db.get(nid)
        if nbrec is not None:
            nbdata = unpack_notebook(nbrec)
            # เก็บค่า is_deleted ถ้ามี (fallback=0)
            is_deleted = nbdata.get('is_deleted', 0)
            packed_nb = pack_notebook(is_deleted,
                                      nbdata['notebook_id'],
                                      nbdata['brand'],
                                      nbdata['serial_num'],
                                      nbdata['rel'],
                                      nbdata['price'],
                                      status)
            nb_db.update(off, packed_nb)   # <-- ใช้ offset แทน id
            log_action(f"Notebook id={nid} status updated to {status} due to sale")
        else:
            print("** Warning: ไม่พบ notebook เพื่ออัปเดตสถานะ **")
    except Exception as e:
        print("** Warning: ไม่สามารถอัปเดตสถานะโน้ตบุ๊กได้:", e)

    log_action(f"Add Soldout id={sid}, nid={nid}, cid={cid}")
```

รูปภาพที่ 4- ฟังก์ชั่น menu_ add_soldout

4.2.9 ฟังก์ชัน update_notebook

ใช้สำหรับแก้ไขข้อมูลโน้ตบุ๊กที่มีอยู่ โดยรับรหัสโน้ตบุ๊กจากผู้ใช้ แล้วดึงข้อมูลเดิมมาแสดง จากนั้นให้ผู้ใช้กรอกข้อมูลใหม่ หากไม่กรอกจะใช้ข้อมูลเดิมแทน แล้วบันทึกข้อมูลใหม่กลับไปยังฐานข้อมูลตรวจสอบว่าข้อมูลมีอยู่หรือไม่ ใช้ unpack_notebook() เพื่อแปลงข้อมูลจากไบต์รับข้อมูลใหม่จากผู้ใช้ พร้อมแสดงค่าปัจจุบันเป็นค่าเริ่มต้นใช้ pack_notebook() เพื่อจัดรูปแบบข้อมูลใหม่ บันทึกข้อมูลด้วย nb_db.update()บันทึกการกระทำด้วย log_action()

```
def update_soldout():
    sid = input_int("ระบุ sold_out_id ที่ต้องการแก้ไข: ", allow_zero=False, positive_only=True)
    offset, rec = so_db.get(sid)
    if rec is None:
        print("** ไม่พบข้อมูล **")
        return
    data = unpack_soldout(rec)
    print("ข้อมูลเดิม:", data)
    nid = input_int(f"notebook_id [{data['notebook_id']}]: ", allow_zero=False, positive_only=True)
    cid = input_int(f"customer_id [{data['customer_id']}]: ", allow_zero=False, positive_only=True)
    name = input_fixed_str(f"ชื่อลูกค้า [{data['name']}]: ", 12) or data['name']
    sold_date = input_fixed_str(f"วันที่ขาย [{data['soldout_date']}]: ", 12) or data['soldout_date']
    st_in = input(f"สถานะ (1/0) [{data['status']}]: ").strip()
    status = int(st_in) if st_in in ('0', '1') else data['status']
    packed = pack_soldout(0, sid, nid, cid, name, sold_date, status)
    so_db.update(sid, packed)

    # อัปเดตสถานะโน้ตบุ๊ก ให้สอดคล้องกับ record การขายนี้ — ใช้ offset ของ notebook
    try:
        off, nbrec = nb_db.get(nid)
        if nbrec is not None:
            nbdata = unpack_notebook(nbrec)
            is_deleted = nbdata.get('is_deleted', 0)
            packed_nb = pack_notebook(is_deleted,
                                      nbdata['notebook_id'],
                                      nbdata['brand'],
                                      nbdata['serial_num'],
                                      nbdata['rel'],
                                      nbdata['price'],
                                      status)
            nb_db.update(off, packed_nb)   # <-- ใช้ offset แทน id
            log_action(f"Notebook id={nid} status updated to {status} due to soldout update")
    except Exception as e:
        print("** Warning: ไม่สามารถอัปเดตสถานะโน้ตบุ๊กได้:", e)

    log_action(f"Update Soldout id={sid}")
```

รูปภาพที่ 4- ฟังก์ชั่น menu_ update_soldout

4.2.10 ฟังก์ชัน delete_notebook

ใช้สำหรับลบข้อมูลโน้ตบุ๊ก โดยรับรหัสโน้ตบุ๊กจากผู้ใช้ แล้วเรียกใช้เมธอด delete() เพื่อลบข้อมูลจากฐานข้อมูลใช้ input_int() รับรหัสโน้ตบุ๊กลบข้อมูลด้วย nb_db.delete()บันทึกการกระทำด้วย log_action()

```
def delete_soldout():
    sid = input_int("ระบุ sold_out_id ที่ต้องการลบ: ", allow_zero=False, positive_only=True)
    so_db.delete(sid)
    log_action(f"Delete SoldOut id={sid}")
```

รูปภาพที่ 4- ฟังก์ชั่น menu_ delete_soldout

4.2.11ฟังก์ชัน view_notebook_menu

ใช้สำหรับแสดงเมนูการดูข้อมูลโน้ตบุ๊ก โดยให้ผู้ใช้เลือกว่าจะดูข้อมูลทั้งหมด หรือกรองตามเงื่อนไข เช่น ยี่ห้อ, สถานะ, ช่วงราคา หรือดูสถิติใช้ print() แสดงเมนูให้ผู้ใช้เลือกใช้ nb_db.iter_active() เพื่อวนลูปข้อมูลโน้ตบุ๊กที่ยังใช้งานอยู่มีตัวเลือกการกรองข้อมูล เช่น:ดูทั้งหมดดูตามยี่ห้อดูตามสถานะ(ขายแล้ว/ยังอยู่ในสต็อก)ดูตามช่วงราคาดูสถิติสินค้าคงคลังและสินค้าที่ขายไป

```
def view_soldout_menu():
    print("\n-- ดูข้อมูลการขาย --")
    print("1) ดูรายการเดียว (by id)")
    print("2) ดูทั้งหมด")
    print("3) ดูแบบกรอง (by วันที่ขาย หรือสถานะ)")
    print("4) สถิติโดยสรุป")
    print("0) กลับเมนูหลัก")
    choice = input("เลือก: ").strip()
    if choice == '1':
        sid = input_int("ระบุ sold_out_id: ", allow_zero=False, positive_only=True)
        offset, rec = so_db.get(sid)
        if rec is None:
            print("** ไม่พบข้อมูล **")
            return
        print(unpack_soldout(rec))
    elif choice == '2':
        for _, rec in so_db.iter_active():
            print(unpack_soldout(rec))
    elif choice == '3':
        print("กรอง: 1=วันที่ขาย, 2=สถานะ")
        g = input("เลือกตัวกรอง: ").strip()
        if g == '1':
            date_str = input("ระบุวันที่ (เช่น 2025-10-01): ").strip()
            for _, rec in so_db.iter_active():
                d = unpack_soldout(rec)
                if d['soldout_date'] == date_str:
                    print(d)
        elif g == '2':
            st = input_status("สถานะที่ต้องการ (1=instock,0=soldout)")
            for _, rec in so_db.iter_active():
                d = unpack_soldout(rec)
                if d['status'] == st:
                    print(d)
    elif choice == '4':
        s = so_db.stats()
        instock = soldout = 0
        for _, rec in so_db.iter_active():
            d = unpack_soldout(rec)
            if d['status'] == 1:
                instock += 1
            else:
                soldout += 1
        print("สรุปการขาย:", s, "| instock=", instock, "soldout=", soldout)
```

รูปภาพที่ 4- ฟังก์ชั่น menu_ view_soldout

4.3 ฟังก์ชัน build_report_text

4.3.1 ฟังก์ชันนี้ใช้สำหรับสร้างรายงานสรุปข้อมูลจากฐานข้อมูลต่าง ๆ ได้แก่ ฐานข้อมูลโน้ตบุ๊ก (nb_db), ลูกค้า (cus_db) และรายการขาย (so_db) โดยจะรวบรวมข้อมูล เช่น จำนวนสินค้าคงคลัง, จำนวนสินค้าที่ขายไป, ราคาสินค้า, และแบรนด์ที่พบ เพื่อใช้ในการแสดงผลหรือวิเคราะห์รูปแบบการเข้ารหัสดังรูปที่ 4-

```python
def build_report_text():
    # เวลา (แสดงออฟเซ็ตโซนเวลา เช่น +07:00)
    now_str = f"{datetime.now().strftime('%Y-%m-%d %H:%M:%S')} ({_tz_offset_str()})"

    # สถิติรวมจากไฟล์
    nb_s = nb_db.stats()

    # สร้างแมปลูกค้า (id -> record) จาก cus_db
    cus_map = {}
    for _, crec in cus_db.iter_active():
        cd = unpack_customer(crec)
        cus_map[cd['customer_id']] = cd

    # สร้างแมป notebook_id -> customer_id จาก so_db (รายการขาย)
    nb_to_cid = {}
    for _, srec in so_db.iter_active():
        sd = unpack_soldout(srec)
        nb_to_cid[sd['notebook_id']] = sd['customer_id']

    # DEBUG: แสดงแม็ปในรายงาน (ลบออกเมื่อเสร็จ)
    debug_lines = []
    debug_lines.append(f"DEBUG: nb_to_cid = {nb_to_cid}")
    debug_lines.append(f"DEBUG: cus_map keys = {list(cus_map.keys())}")

    # รวบรวมข้อมูลสำหรับตาราง และคำนวณสรุป
    nb_rows = []
    prices = []
    brand_counter = Counter()
    stock = sold = 0

    for _, rec in nb_db.iter_active():
        d = unpack_notebook(rec)

        status_txt = 'Active' if d['status'] == 1 else 'Sold Out'
        sold_txt = 'Yes' if d['status'] == 0 else 'No'
        if d['status'] == 1:
            stock += 1
        else:
            sold += 1
```

รูปภาพที่ 4- ฟังก์ชั่น **build_report_text** ใช้ข้อมูลสร้างรายงานสรุป

4.3.2 ฟังก์ชันนี้ยังคงทำหน้าที่ในการสร้างรายงานสรุปข้อมูลจากฐานข้อมูล โดยในส่วนนี้จะเน้นการสรุปสถิติและแสดงกิจกรรมล่าสุดที่เกิดขึ้นในระบบกำหนดหัวตารางสำหรับข้อมูลโน้ตบุ๊ก พร้อมความกว้างของแต่ละคอลัมน์กำหนดการจัดตำแหน่งข้อมูล เช่น ขวา (r) หรือซ้าย (l)สร้างตารางด้วยฟังก์ชัน _render_table() โดยใช้หัวตาราง, ข้อมูล และการจัดตำแหน่งแสดงสรุปจำนวนโน้ตบุ๊กทั้งหมด, ที่ยังใช้งานอยู่, ที่ถูกลบ, ที่ขายไป และที่ยังมีอยู่แสดงสถิติราคาสินค้า เช่น ต่ำสุด, สูงสุด, เฉลี่ยเตรียมพื้นที่สำหรับแสดงจำนวนโน้ตบุ๊กตามแบรนด์ใช้ brand_counter ซึ่งเป็น Counter สำหรับนับจำนวนโน้ตบุ๊กแต่ละแบรนด์หากมีข้อมูล จะแสดงจำนวนของแต่ละแบรนด์หากไม่มีข้อมูล จะแสดงข้อความว่าไม่มีข้อมูสร้างส่วนแสดงกิจกรรมล่าสุดที่เกิดขึ้นในระบบ เช่น การเพิ่ม, ลบ, หรือแก้ไขข้อมูลแสดงรายการล่าสุดไม่เกิน 50 รายการหากไม่มีข้อมูลกิจกรรม จะแสดงข้อความว่าไม่มี

```python
def build_report_text():
    # ตารางหลัก (NotebookID ตามด้วย CusID, Tel, Address)
    nb_headers = [
        ("NotebookID", 12),
        ("CusID", 8),
        ("Tel", 12),
        ("Address", 24),
        ("Brand", 12),
        ("Serial", 16),
        ("Year", 6),
        ("Price (THB)", 12),
        ("Status", 10),
        ("Sold", 6),
    ]
    nb_aligns = ['r', 'r', 'l', 'l', 'l', 'l', 'r', 'r', 'l', 'l']
    nb_table = _render_table(nb_headers, nb_rows, nb_aligns)

    # สรุป (เฉพาะ Active)
    summary_lines = [
        "",
        "Summary (เฉพาะสถานะ Active)",
        f"  Total Notebooks (records): {nb_s['total_slots']}",
        f"  Active Notebooks: {nb_s['active']}",
        f"  Deleted Notebooks: {nb_s['deleted']}",
        f"  Currently Sold: {sold}",
        f"  Available Now: {stock}",
        "",
        "Price Statistics (THB, Active only):",
        f"  Min : {p_min:.2f}" if p_min is not None else "  Min : N/A",
        f"  Max : {p_max:.2f}" if p_max is not None else "  Max : N/A",
        f"  Avg : {p_avg:.2f}" if p_avg is not None else "  Avg : N/A",
        "",
        "Notebooks by Brand (Active only):",
    ]
    if brand_counter:
        for brand, cnt in sorted(brand_counter.items()):
            summary_lines.append(f"  {brand} : {cnt}")
    else:
        summary_lines.append("  (none)")

    # กิจกรรมล่าสุด
    activity_block = ["", "Recent Activities:"]
    last_n = activity_log[-50:]
    if last_n:
        activity_block.extend(last_n)
    else:
        activity_block.append("(no activities in this session)")

    parts = []
    parts.extend(header_lines)
    parts.append(nb_table)
    parts.extend(summary_lines)
    parts.extend(activity_block)
    parts.append("")
    parts.extend(debug_lines)    # <-- ให้บล็อก debug แสดงในท้ายรายงาน
    parts.append("")

    return "\n".join(parts)
```

รูปภาพที่ 4- ฟังก์ชั่น **build_report_text** ใช้ข้อมูลสร้างรายงานสรุป

4.4 main_menu ระบบซื้อขายแลปท็อป

4.4.1 ฟังก์ชันนี้ใช้สำหรับแสดงเมนูหลักของโปรแกรมให้ผู้ใช้เลือกดำเนินการต่าง ๆ เช่น เพิ่มข้อมูล, แก้ไข, ลบ, ดูข้อมูล, สร้างรายงาน แสดงเมนูหลักให้ผู้ใช้เลือก รับค่าตัวเลือกจากผู้ใช้ผ่าน input() และลบช่องว่างด้วย .strip() เมนูเพิ่มข้อมูล (Add)ให้ผู้ใช้เลือกว่าจะเพิ่มข้อมูลประเภทใด เรียกใช้ฟังก์ชันที่เกี่ยวข้องตามประเภทที่เลือก เมนูแก้ไขข้อมูล (Update) ให้ผู้ใช้เลือกว่าจะแก้ไขข้อมูลประเภทใด เรียกใช้ฟังก์ชันที่เกี่ยวข้องตามประเภทที่เลือกเมนูลบข้อมูล (Delete) ห้ผู้ใช้เลือกว่าจะลบข้อมูลประเภทใด เมนูดูข้อมูล (View) ให้ผู้ใช้เลือกว่าจะดูข้อมูลประเภทใด เมนูสร้างรายงาน (.txt) สร้างรายงานสรุปข้อมูลโดยเรียกใช้ build_report_text() ออกจากโปรแกรม

```
def main_menu():
        print("1) Add (เพิ่ม)")
        print("2) Update (แก้ไข)")
        print("3) Delete (ลบ)")
        print("4) View (ดู)")
        print("5) Report (.txt) (สร้างรายงาน)")
        print("0) Exit")
        choice = input("เลือกเมนู: ").strip()

        if choice == '1':
            print("\n-- Add --")
            print("1) Customer")
            print("2) Notebook")
            print("3) SoldOut")
            print("0) กลับ")
            c = input("เลือก: ").strip()
            if c == '1': add_customer()
            elif c == '2': add_notebook()
            elif c == '3': add_soldout()

        elif choice == '2':
            print("\n-- Update --")
            print("1) Customer")
            print("2) Notebook")
            print("3) SoldOut")
            print("0) กลับ")
            c = input("เลือก: ").strip()
            if c == '1': update_customer()
            elif c == '2': update_notebook()
            elif c == '3': update_soldout()

        elif choice == '3':
            print("\n-- Delete --")
            print("1) Customer")
            print("2) Notebook")
            print("3) SoldOut")
            print("0) กลับ")
            c = input("เลือก: ").strip()
            if c == '1': delete_customer()
            elif c == '2': delete_notebook()
            elif c == '3': delete_soldout()

        elif choice == '4':
            print("\n-- View --")
            print("1) Customer")
            print("2) Notebook")
            print("3) SoldOut")
            print("0) กลับ")
            c = input("เลือก: ").strip()
            if c == '1': view_customer_menu()
            elif c == '2': view_notebook_menu()
            elif c == '3': view_soldout_menu()

        elif choice == '5':
            report_text = build_report_text()
            with open(REPORT_FILE, 'w', encoding='utf-8') as rf:
                rf.write(report_text)
            log_action(f"Report written: {REPORT_FILE}")
            print(f"Report saved to {REPORT_FILE}")

        elif choice == '0':
            print("ลาก่อน")
            break

        else:
            print("** เมนูไม่ถูกต้อง กรุณาลองใหม่ **")

if __name__ == '__main__':
    main_menu()
```

รูปภาพที่ 4- main_menu

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงงานนี้มีวัตถุประสงค์เพื่อพัฒนาระบบจัดการข้อมูลร้านขายโน้ตบุ๊ก โดยสามารถเพิ่ม แก้ไข ลบ ดูข้อมูล และสร้างรายงานสรุปได้อย่างมีประสิทธิภาพ ระบบถูกออกแบบให้รองรับข้อมูลลูกค้า ข้อมูลสินค้า และข้อมูลการขาย พร้อมทั้งมีการจัดเก็บข้อมูลในรูปแบบไบต์เพื่อความรวดเร็วและประหยัดพื้นที่ผลการดำเนินงานแสดงให้เห็นว่า:ระบบสามารถทำงานได้ครบตามฟังก์ชันที่กำหนด มีการจัดการข้อมูลอย่างเป็นระบบสามารถสร้างรายงานสรุปในรูปแบบ .txt ได้อย่างถูกต้องมีเมนูใช้งานที่เข้าใจง่ายและเหมาะสมกับผู้ใช้ทั่วไป

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

ระหว่างการพัฒนาโครงงานพบปัญหาและอุปสรรคหลายประการ เช่น:การจัดการข้อมูลแบบไบต์ต้องใช้ความเข้าใจเรื่อง struct และการจัดรูปแบบข้อมูลอย่างละเอียดการเชื่อมโยงข้อมูลระหว่างลูกค้า โน้ตบุ๊ก และรายการขายต้องมีการควบคุมความสัมพันธ์อย่างรัดกุมการตรวจสอบความถูกต้องของข้อมูลที่รับจากผู้ใช้ต้องมีการจัดการข้อผิดพลาดอย่างรอบคอบการแสดงผลรายงานต้องมีการจัดรูปแบบตารางให้เหมาะสมกับการอ่าน

5.3 ข้อเสนอแนะ

พื่อพัฒนาโครงงานให้ดียิ่งขึ้นในอนาคต มีข้อเสนอแนะดังนี้:ควรเพิ่มระบบฐานข้อมูลแบบ SQL หรือ NoSQL เพื่อรองรับข้อมูลจำนวนมากและการค้นหาที่รวดเร็วเพิ่มระบบตรวจสอบความถูกต้องของข้อมูลอัตโนมัติ เช่น การตรวจสอบหมายเลขโทรศัพท์หรือวันที่พัฒนาอินเทอร์เฟซแบบกราฟิก (GUI) เพื่อให้ผู้ใช้ทั่วไปสามารถใช้งานได้ง่ายขึ้นเพิ่มระบบสำรองข้อมูลและการกู้คืนข้อมูลเพื่อความปลอดภัย

5.4 สิ่งที่ผู้จัดทำได้รับในการพัฒนาโครงงาน

จากการดำเนินโครงงานนี้ ผู้จัดทำได้รับประสบการณ์และความรู้ในหลายด้าน ได้แก่: ความเข้าใจในการจัดการข้อมูลแบบไบต์และการใช้โมดูล struct การออกแบบระบบที่มีความสัมพันธ์ระหว่างข้อมูลหลายประเภทการเขียนโปรแกรมเชิงโครงสร้างและการจัดการข้อผิดพลาดการสร้างรายงานและการจัดรูปแบบข้อมูลให้อ่านง่ายการวางแผนและแก้ไขปัญหาอย่างเป็นระบบ