

ระบบยืม – คืนหนังสือห้องสมุด
Borrow and return library books

จัดทำโดย

นายธีรภัทร มังเทศ 6806022510238 sec2

นายลัทธพล ศิริหาล้า 6806022510289 sec2

โครงการนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2568
ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

คำนำ

โครงการเรื่อง ระบบยืม-คืนหนังสือห้องสมุด (Borrow and Return Library Books) จัดทำขึ้นเพื่อพัฒนาและปรับปรุงกระบวนการยืม-คืนหนังสือของห้องสมุดให้มีความสะดวก รวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้น โดยเป็นส่วนหนึ่งของการศึกษาในสาขาวิศวกรรมสารสนเทศและเครือข่ายภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ทั้งนี้ผู้จัดทำหวังว่าโครงการดังกล่าวจะเป็นประโยชน์ต่อการเรียนรู้และการนำไปประยุกต์ใช้จริงในการพัฒนาระบบสารสนเทศของห้องสมุดต่อไป

สารบัญ

บทที่	หน้า
คำนำ	ก
สารบัญ	ข
สารบัญรูปภาพ	ง
สารบัญรูปภาพ(ต่อ)	จ
สารบัญรูปภาพ(ต่อ)	ฉ
สารบัญตาราง	ช
บทที่ 1	1
1.1 วัตถุประสงค์ของโครงการ	1
1.2 ขอบเขตของโครงการ	1
1.3 ประโยชน์ที่ได้รับ	2
1.4 เครื่องมือที่คาดว่าจะต้องใช้	2
บทที่ 2	3
2.1 ภาษา Python	3
2.2 ไฟล์ไบนารี (Binary File)	3
2.3 โมดูล struct	3
2.4 การเขียนโปรแกรมแบบ CRUD	3
2.5 ระบบจัดการห้องสมุด (Library Management System)	4
บทที่ 3	7
3.1 การเริ่มต้นโปรแกรม	7
3.2 เมนูหลัก	7
3.3 การจัดการหนังสือ	7
3.4 การจัดการสมาชิก	11
3.5 การยืมและคืนหนังสือ	14
3.6 การสร้างรายงาน Generate Report (.txt)	16

สารบัญ (ต่อ)

บทที่	หน้า
3.6 ดูมุมมอง	17
3.7 การออกจากโปรแกรม	23
บทที่ 4	14
4.1 โมดูลที่นำเข้าไปในระบบ Library Manager	14
4.2 อธิบายการทำงานโค้ด	14
บทที่ 5	33
5.1 สรุปผลการดำเนินงาน	33
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	33
5.3 ข้อเสนอแนะ	33
5.4 ประโยชน์ที่ผู้จัดทำได้รับ	33

สารบัญรูปภาพ

รูปภาพที่	หน้า
รูปภาพ 3-1 รันโปรแกรมด้วยคำสั่ง	7
รูปภาพ 3-2 เมนูหลัก	7
รูปภาพ 3-3 เลือก 1) Book	8
รูปภาพ 3-4 1) Add Book	8
รูปภาพ 3-5 กรอกข้อมูลเพื่อเพิ่ม	8
รูปภาพ 3-6 ยืนยันข้อมูล	8
รูปภาพ 3-9 2) Book	9
รูปภาพ 3-10 2) Update Book	9
รูปภาพ 3-11 กรอกแก้ไขข้อมูลและยืนยัน	10
รูปภาพ 3-12 2) Book	10
รูปภาพ 3-13 ลบหนังสือ	11
รูปภาพ 3-14 กรอกไอดีหนังสือเพื่อลบ	11
รูปภาพ 3-15 เลือก 2) Member	11
รูปภาพ 3-16 เลือก 1) Add Member	12
รูปภาพ 3-17 กรอกข้อมูลเพื่อเพิ่มผู้ใช้	12
รูปภาพ 3-18 เลือก 2) Member	12
รูปภาพ 3-19 เลือก 2) Update Member	13
รูปภาพ 3-20 กรอกไอดีและกรอกข้อมูลเพื่อแก้ไขผู้ใช้	13
รูปภาพ 3-21 เลือก 2) Member	13
รูปภาพ 3-22 เลือก 3) Delete Member	14
รูปภาพ 3-23 กรอกไอดีเพื่อลบข้อมูล	14
รูปภาพ 3-24 เลือก 3) Borrow Book	14
รูปภาพ 3-25 กรอกไอดีหนังสือ	15
รูปภาพ 3-26 กรอกผู้ใช้ไอดี	15

สารบัญรูปภาพ(ต่อ)

รูปภาพที่	หน้า
รูปภาพ 3-27 ยืนยันข้อมูล	15
รูปภาพ 3-28 กรอกไอดีเพื่อลบผู้ใช้งาน	16
รูปภาพ 3-29 กรอกไอดีเพื่อลบผู้ใช้งาน	16
รูปภาพ 3-30 ยืนยันเพื่อคืนหนังสือ	16
รูปภาพ 3-31 สร้างรายงาน	17
รูปภาพ 3-32 รายงาน Report.txt	17
รูปภาพ 3-33 5) View	18
รูปภาพ 3-34 View_all_book	18
รูปภาพ 3-35 View_all_Book	19
รูปภาพ 3-36 View_all_Loan	19
รูปภาพ 3-37 View_Filter	20
รูปภาพ 3-38 View_filter_book	20
รูปภาพ 3-39 View_filter_member	21
รูปภาพ 3-39 View_Single	21
รูปภาพ 3-40 View_Single_Book	22
รูปภาพ 3-41 View_Single_member	22
รูปภาพ 3-42 View_Single_loan	23
รูปภาพ 3-21 0) Exit	23
รูปภาพ 3-21 ออกโปรแกรม	23
รูปภาพ 4-1 main_menu()	15
รูปภาพ 4-2 main_menu() ต่อ	16
รูปภาพ 4-3 add_book()	17
รูปภาพ 4-4 add_member()	18
รูปภาพ 4-5 add_loan()	19

สารบัญรูปภาพ(ต่อ)

รูปภาพที่	หน้า
รูปภาพ 4-6 add_loan() ต่อ 1	20
รูปภาพ 4-7 add_loan() ต่อ 2	20
รูปภาพ 4-8 return_loan()	21
รูปภาพ 4-9 return_loan() ต่อ	22
รูปภาพ 4-10 generate_report()	23
รูปภาพ 4-11 generate_report() ต่อ	23
รูปภาพ 4-12 generate_report() ต่อ	24
รูปภาพ 4-13 update_book()	25
รูปภาพ 4-14 update_book() ต่อ	25
รูปภาพ 4-15 delete_book()	26
รูปภาพ 4-16 delete_book() ต่อ	27
รูปภาพ 4-17 Update_member()	28
รูปภาพ 4-18 Update_member()	28
รูปภาพ 4-19 delete_member()	29
รูปภาพ 4-20 view()	31
รูปภาพ 4-21 view() ต่อ 1	31
รูปภาพ 4-22 view() ต่อ 2	32
รูปภาพ 4-23 view() ต่อ 3	32

สารบัญตาราง

ตารางที่	หน้า
ตาราง 2.1 ตาราง books	5
ตาราง 2.2 ตาราง members	6
ตาราง 2.3 ตาราง Borrow	7

บทที่ 1

บทนำ

1.1 วัตถุประสงค์ของโครงการ

- 1.1.1 เพื่อพัฒนาระบบยืม – คืนหนังสือห้องสมุดได้อย่างมีประสิทธิภาพ
- 1.1.2 เพื่อฝึกฝนทักษะการเขียนโปรแกรมด้วย Python
- 1.1.3 เพื่อเรียนรู้วิธีการจัดการข้อมูลและไฟล์
- 1.1.4 เพื่อเรียนรู้การทำงานร่วมกันเป็นทีม

1.2 ขอบเขตของโครงการ

- 1.2.1 ระบบยืม – คืนหนังสือห้องสมุดมีฟังก์ชันพื้นฐานทั้งหมด 15 ฟังก์ชัน เช่น
 - 1.เพิ่มหนังสือ
 - 2.แก้ไขหนังสือ
 - 3.ดูข้อมูลหนังสือ
 - 4.ลบหนังสือ
 - 5.กลับไปเมนู
 - 6.เพิ่มสมาชิก
 - 7.ลบสมาชิก
 - 8.แก้ไขสมาชิก
 - 9.แสดงสมาชิกทั้งหมด
 - 10.ยืมหนังสือ
 - 11.คืนหนังสือ
 - 12.แสดงข้อมูลการยืม
 - 13.แสดงข้อมูลการยืมปัจจุบัน
 - 14.เมนูกลางระบบการยืม - คืนหนังสือห้องสมุด
 - 15.เมนูออกจากหน้าปัจจุบัน
- 1.2.2 ระบบยืม – คืนหนังสือห้องสมุดประกอบด้วย 7 필ด์ ได้แก่
 - 1.Book ID
 - 2.Title
 - 3.Author
 - 4.Year

5.Copies

6.Borrowed By

7.Status

1.2.3 ระบบยืม - คืบหนังสือห้องสมุดมีการจัดเก็บข้อมูลหนังสือไว้ใน Text File ชื่อ report ซึ่งมี รหัสหนังสือ ชื่อหนังสือ ชื่อผู้เขียน ปีที่เขียน ชื่อผู้ยืม จำนวนหนังสือทั้งหมด รายการผู้ยืม สถานะการยืม จำนวนหนังสือที่ถูหายืม จำนวนหนังสือที่เหลือให้ยืม สถิติหนังสือในการยืม

1.2.4 ระบบยืม - คืบหนังสือห้องสมุดจะมีเมนูเพื่อให้ผู้ใช้สามารถเลือก ดำเนินการได้

1.3 ประโยชน์ที่ได้รับ

1.3.1 พัฒนาระบบที่สามารถทำการยืม - คืบหนังสือได้อย่างมีประสิทธิภาพ

1.3.2 พัฒนาทักษะการเขียนโปรแกรม

1.3.3 เรียนรู้การจัดการข้อมูลและไฟล์

1.3.4 เรียนรู้การทำงานร่วมกันเป็นทีม

1.4 เครื่องมือที่คาดว่าจะต้องใช้

1.4.1 ภาษา Python

1.4.2 Visualstudio

1.4.3 Microsoft Office

บทที่ 2

เอกสารที่เกี่ยวข้อง

2.1 ภาษา Python

Python เป็นภาษาระดับสูงที่มีโครงสร้างภาษาที่อ่านง่ายและใช้งานง่าย เหมาะสำหรับผู้เริ่มต้นและนักพัฒนาระดับมืออาชีพ มีจุดเด่นคือเป็นภาษาที่สามารถทำงานได้หลากหลาย ทั้งด้านงานวิทยาศาสตร์ ข้อมูล ปัญญาประดิษฐ์ และการพัฒนาโปรแกรมประยุกต์ทั่วไป

2.2 ไฟล์ไบนารี (Binary File)

2.2.1 ความหมายของไฟล์ไบนารี

ไฟล์ไบนารี คือ ไฟล์ที่เก็บข้อมูลในรูปแบบตัวเลขฐานสอง (0 และ 1) ซึ่งไม่สามารถอ่านได้โดยตรงเหมือนไฟล์ข้อความ (Text File)

2.2.2 การจัดการไฟล์ไบนารีใน Python

ในภาษา Python สามารถเปิดไฟล์ไบนารีได้โดยใช้โหมดต่าง ๆ เช่น "rb", "wb", "ab", "rb+"

2.3 โมดูล struct

2.3.1 ความหมายของ Struct

โมดูล struct ใช้สำหรับแปลงข้อมูลของ Python เช่น int, string, float ให้เป็นข้อมูลแบบ bytes เพื่อนำไปเก็บในไฟล์ไบนารี

2.3.2 การใช้งาน Struct

- struct.pack(fmt, data...) แปลงข้อมูลเป็น bytes
- struct.unpack(fmt, bytes_data) แปลง bytes กลับเป็นข้อมูล

2.4 การเขียนโปรแกรมแบบ CRUD

2.4.1 ความหมายของ CRUD

CRUD ย่อมาจาก Create, Read, Update, Delete ซึ่งเป็นหลักการพื้นฐานของการจัดการข้อมูล

2.4.2 การประยุกต์ใช้ CRUD ในโครงการ

- Create เพิ่มข้อมูลใหม่ เช่น เพิ่มหนังสือหรือสมาชิก
- Read อ่านข้อมูล เช่น แสดงรายการหนังสือ
- Update แก้ไขข้อมูล เช่น อัปเดตจำนวนเล่มคงเหลือ

- Delete ลบข้อมูล โดยในโครงงานนี้ใช้วิธี Soft delete คือไม่ลบจริง แต่ตั้งค่า Active=0

2.5 ระบบจัดการห้องสมุด (Library Management System)

2.5.1 ความหมายของระบบจัดการห้องสมุด

ระบบจัดการห้องสมุด คือ โปรแกรมที่พัฒนาขึ้นเพื่อช่วยในการจัดเก็บข้อมูลหนังสือ ข้อมูลสมาชิก และการยืม-คืนหนังสือ

2.5.2 โปรแกรม Library Manager

2.5.2.1 วัตถุประสงค์

เพื่อสร้างระบบจัดการห้องสมุด (Library Manager) ที่สามารถทำงาน เพิ่ม-แก้ไข-ลบ-ค้นหา ได้ทั้งหนังสือ (Books), สมาชิก (Members) และการยืม-คืน (Loans) เก็บข้อมูลแบบ ไฟล์ไบนารี (.bin) โดยใช้ struct.pack / struct.unpack เพื่อควบคุมโครงสร้างข้อมูล

- มีการ Soft Delete (ลบโดยไม่ลบจริง) เพื่อรักษาประวัติข้อมูล
- สร้าง รายงาน (report.txt) สรุปจำนวนระเบียบและกิจกรรมล่าสุด

2.5.2.2 โครงสร้างไฟล์ข้อมูล

- books.bin – เก็บข้อมูลหนังสือ
- members.bin – เก็บข้อมูลสมาชิก
- loans.bin – เก็บข้อมูลการยืม-คืน
- report.txt – เก็บรายงานสรุปผล

2.5.2.3 โครงสร้างข้อมูล (Struct Format)

2.5.2.3.1 books

- id ฟิลด์นี้ใช้เป็นรหัสเฉพาะของหนังสือแต่ละเล่มในระบบ โดยกำหนดให้ไม่ซ้ำกัน เพื่อให้สามารถอ้างอิงหรือเชื่อมโยงไปยังข้อมูลอื่นได้ เช่น ตอนสร้างประวัติการยืมจะใช้ book_id มาจับคู่กับ id ของหนังสือ ทำให้แม้จะมีหนังสือชื่อเหมือนกันแต่ต่างเล่มก็แยกออกจากกันได้อย่างชัดเจน
- title ฟิลด์นี้เก็บชื่อหนังสือ ซึ่งเป็นข้อมูลหลักที่สมาชิกใช้ค้นหา เช่น การพิมพ์ค้นหาหนังสือในระบบ หรือแสดงผลในรายงานการยืมคืน หากไม่มีชื่อก็จะไม่สามารถบอกได้ว่าหนังสือเล่มไหนถูกยืมหรือเหลืออยู่
- author ฟิลด์นี้เก็บชื่อผู้แต่งเพื่อใช้เป็นอีกหนึ่งเงื่อนไขการค้นหา หรืออาจใช้ในการจัดหมวดหมู่ เช่น รวมผลงานของนักเขียนเดียวกัน และยังช่วยเพิ่มความถูกต้องเวลาตรวจสอบว่าสมาชิกต้องการหนังสือเรื่องเดียวกันหรือแค่คล้ายกัน
- year ฟิลด์นี้เก็บปีที่หนังสือถูกตีพิมพ์ ข้อมูลนี้มีประโยชน์ในการจัดเรียงหนังสือ เช่น การหาว่าหนังสือใหม่หรือเก่า รวมถึงใช้ในการรายงานอายุของหนังสือว่ามีหนังสือรุ่นใหม่กี่ตัว

จัดหามาทดแทนหรือไม่

- total ฟิลด์นี้บันทึกจำนวนเล่มทั้งหมดที่ห้องสมุดมีในสต็อก เพื่อใช้ควบคุมจำนวนจริงว่าเรามีหนังสือกี่เล่มสำหรับบริการสมาชิก ไม่ว่าจะเป็นการยืม การคืน หรือการเพิ่มหนังสือใหม่
- available ฟิลด์นี้เก็บจำนวนเล่มที่ยังสามารถยืมได้จริงในปัจจุบัน โดยค่าจะเปลี่ยนไปเมื่อมีการยืมหรือคืน เช่น ถ้าสมาชิกยืมไปหนึ่งเล่ม available ก็จะลดลง และถ้ามีการคืนค่าก็จะเพิ่มขึ้น
- active ฟิลด์นี้ใช้ระบุสถานะว่าหนังสือยังอยู่ในระบบหรือไม่ เช่น ค่า 1 หมายถึงยังเปิดให้ยืมได้ ส่วนค่า 0 หมายถึงเลิกใช้งานแล้ว แต่ยังเก็บข้อมูลไว้เพื่อการอ้างอิงภายหลังโดยไม่ลบออกจริง
- last_modified ฟิลด์นี้เก็บเวลาที่มีการแก้ไขข้อมูลครั้งล่าสุด เช่น การเพิ่มเล่มใหม่ การแก้ไข การอัปเดตจำนวนเล่มหรือสถานะ เพื่อใช้ในการตรวจสอบย้อนหลังหรือทำรายงานว่ามี การปรับปรุงข้อมูลเมื่อไร

Field	Type	Description	Example
ID	int	เลขประจำเล่มหนังสือ	1001
Title	string	ชื่อหนังสือ	"Computer"
Author	string	ชื่อผู้แต่ง	"Tanenbaum"
Year	int	ปีพิมพ์	2021
Total	int	จำนวนเล่มทั้งหมด	5
Available	int	จำนวนเล่มว่างให้ยืม	3
Active	bool/int	สถานะใช้งาน (1=Active, 0=Soft Delete)	1
last_modified	timestamp	เวลาแก้ไขล่าสุด	1694851200

ตาราง 2.1 ตาราง books

2.5.2.3.1 members

- id ฟิลด์นี้ใช้เป็นตัวระบุเฉพาะของสมาชิกแต่ละคน เพื่อป้องกันความซ้ำซ้อนและใช้ในการเชื่อมโยงกับข้อมูลการยืม เช่น ใน Loan จะเก็บ member_id เพื่อบอกว่าใครเป็นผู้ยืม
- name ฟิลด์นี้เก็บชื่อจริงของสมาชิกเพื่อใช้ในการระบุตัวตนและแสดงผล เช่น รายงานผู้ยืมหนังสือ รายการแจ้งเตือน หรือการค้นหาสมาชิกในระบบ
- phone ฟิลด์นี้ใช้เก็บเบอร์โทรศัพท์ของสมาชิก เพื่อให้เจ้าหน้าที่สามารถติดต่อได้ เช่น โทรแจ้งเมื่อมีการยืมหนังสือเกินกำหนดคืน
- active ฟิลด์นี้เก็บสถานะของสมาชิก เช่น 1 หมายถึงยังใช้งานได้ สามารถยืมคืนหนังสือได้ ส่วน 0 หมายถึงบัญชีถูกปิดหรือพักการใช้งาน แต่ยังเก็บข้อมูลไว้เพื่อการอ้างอิงย้อนหลัง
- last_modified ฟิลด์นี้เก็บเวลาที่มีการแก้ไขข้อมูลสมาชิกครั้งล่าสุด เช่น การเปลี่ยนชื่อ การอัปเดตเบอร์โทรศัพท์ หรือการปรับสถานะ active เพื่อความถูกต้องของประวัติ

Field	Type	Description	Example
ID	int	เลขประจำตัวสมาชิก	501
Name	string	ชื่อสมาชิก	"Somchai"
Phone	string	เบอร์โทร	"0812345678"
Address	string	ที่อยู่	"Bangkok"
Active	bool/int	สถานะสมาชิก (1=Active, 0=Soft Delete)	1
last_modified	timestamp	เวลาแก้ไขล่าสุด	1694851300

ตาราง 2.2 ตาราง members

2.5.2.3.1 members

- id ฟิลด์นี้เป็นรหัสเฉพาะของการยืมแต่ละครั้ง ใช้เพื่อแยกออกจากกันว่าเป็นการยืมใด แม้สมาชิกคนเดียวกันจะยืมหลายเล่มหรือหลายครั้งก็ยังแยกได้ชัดเจน
- book_id ฟิลด์นี้เชื่อมโยงกับตาราง Book โดยเก็บรหัสหนังสือที่ถูกยืม เพื่อให้ระบบรู้ว่าการยืมครั้งนี้เกี่ยวข้องกับหนังสือเล่มไหน
- member_id ฟิลด์นี้เชื่อมโยงกับตาราง Member โดยเก็บรหัสสมาชิกผู้ยืม เพื่อให้ระบบรู้ว่าหนังสือเล่มนี้อยู่ในความรับผิดชอบของใคร
- borrow_date ฟิลด์นี้เก็บวันเวลาที่เริ่มยืมหนังสือ ข้อมูลนี้ใช้ตรวจสอบกำหนดคืน เช่น กำหนดให้คืนภายใน 7 วัน และยังใช้ในการคำนวณค่าปรับหากเกินกำหนด
- return_date ฟิลด์นี้เก็บวันเวลาที่หนังสือถูกคืน หากยังไม่คืนก็อาจปล่อยค่านี้อย่างว่างหรือตั้งค่า

เป็นศูนย์ เพื่อใช้ตรวจสอบว่ายังมีหนังสือที่ค้างยืมอยู่หรือไม่

- active ฟิลด์นี้ใช้ระบุว่าสถานะการยืมยังคงใช้งานอยู่หรือไม่ เช่น 1 หมายถึงยังยืมอยู่ (ยังไม่คืน) ส่วน 0 หมายถึงยืมเสร็จสิ้นแล้วหรือข้อมูลนั้นถูกยกเลิก
- last_modified ฟิลด์นี้เก็บเวลาที่ข้อมูลการยืมครั้งนี้ถูกแก้ไขล่าสุด เช่น การบันทึกการคืนหนังสือหรือการเปลี่ยนสถานะ active เพื่อช่วยติดตามการอัปเดตย้อนหลัง

Field	Type	Description	Example
Loan_ID	int	เลขประจำตัวรายการยืม	2001
Book_ID	int	ชี้ไปยังหนังสือ (FK → Book.ID)	1001
Member_ID	int	ชี้ไปยังสมาชิก (FK → Member.ID)	501
borrow_date	timestamp	วันที่ยืม	1694851400
return_date	timestamp	วันที่คืน (0 ถ้ายังไม่คืน)	0
Active	bool/int	สถานะรายการยืม (1=Active, 0=Inactive)	1
last_modified	timestamp	เวลาแก้ไขล่าสุด	1694851500

ตาราง 2.3 ตาราง Borrow

2.5.2.4 เมนูหลักของโปรแกรม

- Add (เพิ่มข้อมูล)
- Update (แก้ไขข้อมูล)
- Delete (ลบข้อมูลแบบ Soft Delete)
- View (ดูข้อมูล)
- Generate Report (.txt)
- Exit (ออกจากโปรแกรม)

2.5.2.5 ตัวอย่างการทำงาน

- Add Book เพิ่มหนังสือใหม่ พร้อมกำหนดจำนวนเล่ม
- Add Member เพิ่มข้อมูลสมาชิกพร้อมเบอร์ติดต่อ
- Add Loan บันทึกการยืมหนังสือ พร้อมลดจำนวนคงเหลือในคลัง
- Return Loan คืนหนังสือและเพิ่มจำนวนคงเหลือ
- Delete Soft Delete โดยตั้งค่า active=0
- Report สร้างไฟล์ report.txt ที่มีสรุปข้อมูลและกิจกรรมล่าสุด

2.5.2.6 คุณสมบัติเด่น

- ใช้ struct ควบคุมขนาด record ให้คงที่
- มีระบบ Soft Delete เพื่อรักษาประวัติ
- ทุกการแก้ไขจะบันทึกเวลา (last_modified)
- มีการ fsync() เพื่อความปลอดภัยของข้อมูล
- รองรับการ ค้นหา / filter เช่น ค้นหาหนังสือจากชื่อหรือผู้เขียน

บทที่ 3

การใช้งานระบบยืมหนังสือ

3.1 การเริ่มต้นโปรแกรม

- 1.1.1 เปิดเทอร์มินอลหรือคอมมานด์ไลน์
- 1.1.2 รันโปรแกรมด้วยคำสั่ง
- 1.1.3 python library_manager.py
- 1.1.4 โปรแกรมจะสร้างไฟล์เดอร์ data และไฟล์เก็บข้อมูลอัตโนมัติ หากยังไม่มี

```
PS C:\Users\User\OneDrive\Desktop\Topology\New folder> python .\Borrow_Book.py
```

รูปภาพ 3-1 รันโปรแกรมด้วยคำสั่ง

3.2 เมนูหลัก

เมื่อรันโปรแกรมจะเจอเมนูหลัก ผู้ใช้งานสามารถพิมพ์ตัวเลขเพื่อเลือกทำงานแต่ละเมนู

```
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) Borrow Book
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit
Select: █
```

รูปภาพ 3-2 เมนูหลัก

3.3 การจัดการหนังสือ

3.3.1 เพิ่มหนังสือ (Book)

เลือก 1) Book จากนั้นกด 2) AddBook จากนั้น กรอกข้อมูล Title ชื่อหนังสือ Author ชื่อผู้แต่ง Year ปีพิมพ์ Total copies จำนวนเล่มทั้งหมด ระบบจะสร้าง ID ของหนังสืออัตโนมัติ และบันทึกลงฐานข้อมูล และยืนยันกด Y ถ้าไม่ต้องการกด N

```

1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) Borrow Book
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit

Select: 

```

รูปภาพ 3-3 เลือก 1) Book

```

--- Book Menu ---
1) Add Book
2) Update Book
3) Delete Book
0) Back

Select: 

```

รูปภาพ 3-4 1) Add Book

```

-- Add Book --
Title: World Map
Author: Earth
Year (e.g., 2024): 2024
Total copies: 2

```

รูปภาพ 3-5 กรอกข้อมูลเพื่อเพิ่ม

```

Please confirm the book information:
ID      : 6
Title   : World Map
Author  : Earth
Year    : 2024
Total   : 2
Avail.  : 2
Save this book? (y/n): 

```

รูปภาพ 3-6 ยืนยันข้อมูล

3.3.2 แก้ไขหนังสือ (Update Book)

เลือก 1) Book หลังจากนั้นเลือก 2) Update Book โปรแกรมจะแสดงรายการหนังสือให้เลือก Book ID สามารถแก้ไข Title, Author, Year, Total copies หากไม่แก้ไขช่องใด ให้เว้นว่าง ระบบจะเก็บค่าปัจจุบันไว้ และยืนยันกด Y ถ้าไม่ต้องการกด N

```
===== Library Manager =====
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) Borrow Book
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit
```

รูปภาพ 3-9 2) Book

```
--- Book Menu ---
1) Add Book
2) Update Book
3) Delete Book
0) Back
```

รูปภาพ 3-10 2) Update Book

```
-- Update Book --
```

ID	Title	Author	Year	Total	Available
1	eart	dragon	2000	2	0
2	word	world	2004	1	0
3	qoweqw	wqeq	2000	2	1
4	team	king	1999	5	1
5	habib saluba	babubu	2000	2	1
6	World Map	Earth	2024	2	2

Enter Book ID to update: 1
 Leave blank to keep current value.
 Title [eart]: Eawn
 Author [dragon]: peach
 Year [2000]: 2001
 Total copies [2]: 1

Please confirm the updated book information:

```

ID      : 1
Title   : Eawn
Author  : peach
Year    : 2001
Total   : 1
Available: 0
Status  : Active
Save changes? (y/n): y
Book updated successfully.
  
```

รูปภาพ 3-11 กรอกแก้ไขข้อมูลและยืนยัน

3.3.3 ลบหนังสือ (Delete Book)

เลือก 6) Delete Book เลือก Book ID ที่ต้องการลบ การลบเป็น Soft Delete หนังสือยังอยู่ในฐานข้อมูลแต่สถานะเป็น Deleted

```
----- Library Manager -----
```

- 1) Book (Add/Update/Delete)
- 2) Member (Add/Update/Delete)
- 3) Borrow Book
- 4) Return Book
- 5) View
- 6) Generate Report (.txt)
- 0) Exit

Select: 1

รูปภาพ 3-12 2) Book

```

--- Book Menu ---
1) Add Book
2) Update Book
3) Delete Book
0) Back

Select: 3

```

รูปภาพ 3-13 ลบหนังสือ

```

-- Delete Book --
+-----+-----+-----+-----+-----+-----+
| ID | Title                | Author      | Year | Total | Available |
+-----+-----+-----+-----+-----+-----+
| 1  | harrypotter ep2      | JK.rowling  | 2003 | 2      | 2          |
| 3  | English              | Sam         | 1500 | 2      | 2          |
+-----+-----+-----+-----+-----+-----+
Enter Book ID to delete: 3
Book deleted (soft delete).

```

รูปภาพ 3-14 กรอกไอดีหนังสือเพื่อลบ

3.4 การจัดการสมาชิก

3.4.1 เพิ่มสมาชิก (Add Member)

เลือก 2) Add Memberกรอกข้อมูล Full Name Phone Address ระบบจะสร้าง Member ID อัตโนมัติ

```

===== Library Manager =====
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) BORROW BOOK
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit

Select: 2

```

รูปภาพ 3-15 เลือก 2) Member

```

--- Member Menu ---
1) Add Member
2) Update Member
3) Delete Member
0) Back

Select: 1

```

รูปภาพ 3-16 เลือก 1) Add Member

```

-- Add Book --
Title: Franken style
Author: Aw
Year (e.g., 2024): 2000
Total copies: 1

Please confirm the book information:
ID      : 7
Title   : Franken style
Author  : Aw
Year    : 2000
Total   : 1
Avail.  : 1
Save this book? (y/n): y

```

รูปภาพ 3-17 กรอกข้อมูลเพื่อเพิ่มผู้ใช้

3.4.2 แก้ไขสมาชิก (Update Member)

เลือก 7) Update Member โปรแกรมจะแสดงรายการผู้ใช้ให้เลือก Member ID สามารถแก้ไข Name Phone Address หากไม่แก้ไขช่องใด ให้เว้นว่าง ระบบจะเก็บค่าปัจจุบันไว้

```

===== Library Manager =====
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) BORROW BOOK
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit

Select: 2

```

รูปภาพ 3-18 เลือก 2) Member

```

--- Member Menu ---
1) Add Member
2) Update Member
3) Delete Member
0) Back

Select: 2

```

รูปภาพ 3-19 เลือก 2) Update Member

```

-- Update Member --
+-----+-----+-----+-----+-----+
| ID | Name                | Phone    | Address  | Status |
+-----+-----+-----+-----+-----+
| 1  | ratapon sirila      | 204124921 | thailand | Active |
| 2  | theerapat           | 304239432 | thailand | Active |
| 3  | Theerpat mangthsdsdss | 000999   | rydd     | Active |
| 4  | puvadon             | 3042424324 | Thailand | Active |
+-----+-----+-----+-----+-----+
Enter Member ID to update: 4
Leave blank to keep current value.
Name [puvadon]: habif
Phone [3042424324]: 23223232
Address [Thailand]: Thailand bangkok

```

รูปภาพ 3-20 กรอกไอดีและกรอกข้อมูลเพื่อแก้ไขผู้ใช้

3.4.3 ลบสมาชิก (Delete Member)

เลือก 8) Delete Member เลือก Member ID ที่ต้องการลบ การลบเป็น Soft Delete ผู้ใช้ยังอยู่ในฐานข้อมูลแต่สถานะเป็น Deleted

```

===== Library Manager =====
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) Borrow Book
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit

Select: 2

```

รูปภาพ 3-21 เลือก 2) Member

```

--- Member Menu ---
1) Add Member
2) Update Member
3) Delete Member
0) Back
Select: 3

```

รูปภาพ 3-22 เลือก 3) Delete Member

```

-- Delete Member (soft delete) --
+-----+-----+-----+-----+-----+
| ID | Name                | Phone   | Address | Status |
+-----+-----+-----+-----+-----+
| 1 | ratapon sirla       | 204124921 | thailand | Active |
| 2 | theerapat           | 304239432 | thailand | Active |
| 3 | Theerpat mangthsdsdss | 000999   | rydd    | Active |
| 4 | puvadon             | 3042424324 | Thailand | Active |
| 5 | Ratapon Sirla       | 063472904 | Thailand | Active |
+-----+-----+-----+-----+-----+
Member ID to delete: 5
Member Ratapon Sirla soft-deleted successfully.

```

รูปภาพ 3-23 กรอกไอดีเพื่อลบข้อมูล

3.5 การยืมและคืนหนังสือ

3.5.1 ยืมหนังสือ (Borrow Book)

เลือก 3) Borrow Book โปรแกรมจะแสดงรายการ หนังสือที่มีจำนวน available > 0

เลือก Book ID ที่ต้องการยืม โปรแกรมจะแสดงรายการสมาชิกทั้งหมด เลือก Member ID ของผู้ยืม

ระบบจะบันทึกการยืมและลดจำนวน available ของหนังสือลง 1

```

===== Library Manager =====
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) Borrow Book
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit
Select: 3

```

รูปภาพ 3-24 เลือก 3) Borrow Book


```
-- Borrow Book --

Available Books:
ID | Title | Author | Year | Available
1 | Eawn | peach | 2001 | 1
2 | word | worlde | 2004 | 0
4 | team | king | 1999 | 0
5 | habib saluba | babubu | 2000 | 1
6 | World Map | Earth | 2024 | 2
7 | Franken style | Aw | 2000 | 1

Book ID ที่ต้องการยืม: 7
```

รูปภาพ 3-25 กรอกไอดีหนังสือ

```
Members:
ID | Name | Phone
1 | ratapon sirla | 204124921
2 | theerapat | 304239432
3 | Theerpat mangthsdsdss | 000999
4 | habif | 23223232

Member ID ผู้ยืม: 4
```

รูปภาพ 3-26 กรอกผู้ใช้ไอดี

```
Please confirm borrow information:
LoanID : (auto) next ID
Book : habib saluba (ID=5)
Member : habif (ID=4)
Confirm borrow? (y/n): y

Borrow success: habif ยืม habib saluba (LoanID=13)
```

รูปภาพ 3-27 ยืนยันข้อมูล

3.5.2 คืนหนังสือ (Return Book)

เลือก 4) Return Book โปรแกรมจะแสดงรายการ หนังสือที่ยังไม่ถูกคืน เลือก Loan ID ของรายการ ยืมที่ต้องการคืนระบบจะบันทึกวันที่คืนและเพิ่มจำนวน available ของหนังสือ

```

===== Library Manager =====
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) Borrow Book
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit
Select: 4

```

รูปภาพ 3-28 กรอกไอดีเพื่อลบผู้ใช้งาน

```

-- Borrow Book --

Available Books:
ID | Title | Author | Year | Available
1 | Eawn | peach | 2001 | 1
2 | word | worlde | 2004 | 0
4 | team | king | 1999 | 0
5 | habib saluba | babubu | 2000 | 1
6 | World Map | Earth | 2024 | 2
7 | Franken style | Aw | 2000 | 1

Book ID ที่ต้องการยืม: 7

```

รูปภาพ 3-29 กรอกไอดีเพื่อลบผู้ใช้งาน

```

Please confirm borrow information:
LoanID : (auto) next ID
Book : Franken style (ID=7)
Member : habif (ID=4)
Confirm borrow? (y/n): y

Borrow success: habif ยืม Franken style (LoanID=14)

```

รูปภาพ 3-30 ยืนยันเพื่อคืนหนังสือ

3.6 การสร้างรายงาน Generate Report (.txt)

เลือก 7) Generate Report ระบบจะสร้างไฟล์ report.txt ในโฟลเดอร์ data

3.5.1 รายงานประกอบด้วย

3.5.2 ประวัติการยืม-คืนทั้งหมด

3.5.3 สถานะปัจจุบันของหนังสือ (จำนวนทั้งหมด, จำนวนว่าง, ผู้ยืม)

3.5.4 สรุปจำนวนหนังสือ สมาชิก หนังสือยืมปัจจุบัน

3.5.6 หนังสือที่ถูกยืมมากที่สุด

```

===== Library Manager =====
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) Borrow Book
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit

Select: 6

```

รูปภาพ 3-31 สร้างรายงาน

Borrow History						
MemberID	Member Name	Phone	Title	Loan Date	Return Date	Status
1	ratapon sirla	204124921	Eawn; word; qoweqw; qoweqw;	2025-10-03	2025-10-04	Returned
2	theerapat	304239432	Eawn	2025-10-03	2025-10-03	Returned
3	Theerpat mangthsdsdss	000999	team; team; team	2025-10-03	-	Borrowed
4	habif	23223232	habib saluba; team; team; h	2025-10-03	-	Borrowed

Summary (Active Books Only)	
- Total Books	: 7
- Active Books	: 6
- Deleted Books	: 1
- Borrowed Now	: 10
- Available Now	: 4

Borrow Statistics (Active only)	
- Most Borrowed Book	: team (5 times)
- Currently Borrowed	: 10
- Active Members	: 4

Report saved to data\report.txt

รูปภาพ 3-32 รายงาน Report.txt

3.6 ดูมุมมอง

3.6.1 view_all

เลือก 4) View และ เลือกว่าจะดูระหว่าง Book/Member/loan เลือก 1) view all หากเลือกเสร็จจะขึ้นตารางข้อมูลที่ จากนั้นกด Enter เพื่อหยุดการทำงานและกด 0 ฟังก์ชันนี้จะหยุดทำงาน

```

===== Library Manager =====
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) Borrow Book
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit

Select: 5

```

รูปภาพ 3-33 5) View

```

-- View Menu --
0) Back to Main Menu
Type (book/member/loan): book
Mode:
1) View All
2) View Filter
3) View Single
0) Back
Select mode: 1

```

ID	Title	Author	Year	Total	Available	Status
1	Eawn	peach	2001	1	1	Active
2	word	world	2004	1	0	Active
4	team	king	1999	5	0	Active
5	habib saluba	babubu	2000	2	0	Active
6	World Map	Earth	2024	2	2	Active
7	Franken style	Aw	2000	1	1	Active

```

Press Enter to continue...

```

รูปภาพ 3-34 View_all_book

```
Type (book/member/loan): member
Mode:
1) View All
2) View Filter
3) View Single
0) Back
Select mode: 1
```

ID	Name	Phone	Address	Status
1	ratapon sirla	204124921	thailand	Active
2	theerapat	304239432	thailand	Active
3	Theerpat mangthsdsdss	000999	rydd	Active
4	habif	23223232	Thailand bangkok	Active

```
Press Enter to continue...
```

รูปภาพ 3-35 View_all_Book

```
-- View Menu --
0) Back to Main Menu
Type (book/member/loan): loan
Mode:
1) View All
2) View Filter
3) View Single
0) Back
Select mode: 1
```

LoanID	BookID	Title	MemberID	Name	Borrow	Return	Status
1	1	Eawn	1	ratapon sirla	2025-10-03	2025-10-04	Returned
2	2	word	1	ratapon sirla	2025-10-03	-	Borrowed
3	1	Eawn	2	theerapat	2025-10-03	2025-10-03	Returned
4	3	qoweqw	1	ratapon sirla	2025-10-03	-	Borrowed
5	3	qoweqw	1	ratapon sirla	2025-10-03	2025-10-03	Returned
6	4	team	3	Theerpat mangthsdsdss	2025-10-03	-	Borrowed
7	4	team	3	Theerpat mangthsdsdss	2025-10-03	-	Borrowed
8	5	habib saluba	4	habif	2025-10-03	-	Borrowed
9	1	Eawn	1	ratapon sirla	2025-10-03	-	Borrowed
10	4	team	4	habif	2025-10-03	-	Borrowed
11	4	team	4	habif	2025-10-03	-	Borrowed
12	4	team	3	Theerpat mangthsdsdss	2025-10-03	-	Borrowed
13	5	habib saluba	4	habif	2025-10-04	-	Borrowed

```
Press Enter to continue...
```

รูปภาพ 3-36 View_all_Loan

3.6.2 view_filter

4) View ให้เลือกว่า ต้องการดูข้อมูลในส่วนใดระหว่าง Book / Member จากนั้นเลือก 1) View Filter หากเลือกตัวเลือกนี้ ระบบจะให้ผู้ใช้พิมพ์ข้อมูลที่ต้องการค้นหา โดยฟังก์ชันจะคัดกรองเฉพาะข้อมูลที่ตรงตามเงื่อนไขที่กรอกไว้ เมื่อต้องการออกให้กด Enter และ พิมพ์เลข 0 เพื่อหยุดการทำงาน

```

1) View All
2) View Filter
3) View Single
0) Back
Select mode: 0

```

รูปภาพ 3-37 View_Filter

```

Type (book/member/loan): book
Mode:
1) View All
2) View Filter
3) View Single
0) Back
Select mode: 2
Keyword in Title (Enter to skip): En
Keyword in Author (Enter to skip): ea
Year (Enter to skip): 2000
Status (active/deleted/Enter to skip):
+---+-----+-----+-----+-----+
| ID | Title | Author | Year | Avail | Status |
+---+-----+-----+-----+-----+
+---+-----+-----+-----+-----+
Found 0 book(s).

Press Enter to continue...

```

รูปภาพ 3-38 View_filter_book

```
-- View Menu --
0) Back to Main Menu
Type (book/member/loan): member
Mode:
1) View All
2) View Filter
3) View Single
0) Back
Select mode: 2
Keyword in Name (Enter to skip): ratapon
Keyword in Phone (Enter to skip):
Keyword in Address (Enter to skip):
Status (active/deleted/Enter to skip):
+---+-----+-----+-----+-----+
| ID | Name          | Phone    | Address  | Status  |
+---+-----+-----+-----+-----+
| 1  | ratapon sirla | 204124921 | thailand | Active  |
| 5  | Ratapon Sirla | 063472904 | Thailand | Deleted |
+---+-----+-----+-----+-----+
Found 2 member(s).

Press Enter to continue...
```

รูปภาพ 3-39 View_filter_member

3.6.3 view_single

เมื่อเลือกเมนู 4) View ให้เลือกว่า ต้องการดูข้อมูลในส่วนใดระหว่าง Book / Member / Loan จากนั้นเลือก 1) View Single ระบบจะให้ค้นหาข้อมูลโดยการกรอก ID ของรายการที่ต้องการ

```
1) View All
2) View Filter
3) View Single
0) Back
Select mode: 2
```

รูปภาพ 3-39 View_Single

```

Type (book/member/loan): book
Mode:
1) View All
2) View Filter
3) View Single
0) Back
Select mode: 3
Book ID: 1
+-----+-----+
| Field   | Value   |
+-----+-----+
| ID      | 1       |
| Title   | Eawn    |
| Author  | peach   |
| Year    | 2001    |
| Total   | 1       |
| Available | 1      |
| Active  | Yes     |
+-----+-----+
Press Enter to continue...

```

รูปภาพ 3-40 View_Single_Book

```

0) Back to Main Menu
Type (book/member/loan): member
Mode:
1) View All
2) View Filter
3) View Single
0) Back
Select mode: 3
Member ID: 1
+-----+-----+
| Field   | Value   |
+-----+-----+
| ID      | 1       |
| Name    | ratapon sirla |
| Phone   | 204124921 |
| Address | thailand |
| Active  | Yes     |
+-----+-----+
Press Enter to continue...

```

รูปภาพ 3-41 View_Single_member


```
-- View Menu --
0) Back to Main Menu
Type (book/member/loan): loan
Mode:
1) View All
2) View Filter
3) View Single
0) Back
Select mode: 3
Loan ID: 1
```

Field	Value
Loan ID	1
Book ID	1
Member ID	1
Borrow Date	2025-10-03 10:35:39
Return Date	2025-10-04 21:01:54
Active	Yes

```
Press Enter to continue...
```

รูปภาพ 3-42 View_Single_loan

3.7 การออกจากโปรแกรม

เลือก 0) Exit หากกด Ctrl+C ระบบจะจับสัญญาณและสร้าง รายงานสุดท้ายก่อนออก

```
===== Library Manager =====
1) Book (Add/Update/Delete)
2) Member (Add/Update/Delete)
3) Borrow Book
4) Return Book
5) View
6) Generate Report (.txt)
0) Exit
```

รูปภาพ 3-21 0) Exit

Exiting...

รูปภาพ 3-21 ออกโปรแกรม

บทที่ 4

การพัฒนาและการทดสอบระบบ

4.1 โมดูลที่นำเข้าไปในระบบ Library Manager

4.1.1 import os

4.1.1.1 ใช้จัดการระบบไฟล์ เช่น ตรวจสอบไฟล์/โฟลเดอร์, สร้างลบไฟล์, ตรวจสอบ path

4.1.2 import struct

4.1.2.1 ใช้สำหรับ pack/unpack ข้อมูลไบนารี เพื่อเขียนลงไฟล์แบบ fixed-length record

4.1.2.2 จำเป็นสำหรับการเก็บข้อมูล Book, Member, Loan ในไฟล์ .bin

4.1.3 import time

ใช้จัดการ timestamp เช่น time.time() เพื่อเก็บเวลาปัจจุบันเป็นตัวเลข float (จำนวนวินาทีตั้งแต่ epoch)

4.1.4 import datetime

ใช้จัดการวันที่และเวลาแบบ human-readable เช่น แปลง timestamp เป็นวันที่แบบ "YYYY-MM-DD HH:MM:SS"

4.1.5 import textwrap

ใช้จัดรูปแบบข้อความ multiline ให้สวยงาม เช่น เมนู main_menu(), add_menu() ใช้ textwrap.dedent() เพื่อลบ indent ของ string

4.1.6 import shutil

ใช้จัดการไฟล์/โฟลเดอร์ขั้นสูง เช่น copy, move, delete folder ที่มีหลายไฟล์

4.2 อธิบายการทำงานโค้ด

4.2.1 main_menu()

โค้ดนี้เป็นฟังก์ชัน main_menu() ที่เป็นเมนูหลักของโปรแกรม Library Manager และทำงานในลูปจนกว่าผู้ใช้จะเลือกออกจากโปรแกรมบรรทัด print(textwrap.dedent(...)) ใช้พิมพ์เมนูแบบหลายบรรทัดโดยเอาช่องว่างนำหน้าที่เกิดจากการเยื้อง (indentation) ออก เพื่อให้เมนูแสดงชิดซ้ายสวยงามโปรแกรมรอรับอินพุตจากผู้ใช้ด้วย input("Select: ").strip() — strip() ตัดช่องว่างทั้งสอง

ด้านออกก่อนประมวลผล เช็คด้วย `s.isdigit()` ว่าผู้ใช้กรอกตัวเลขอย่างเดียวหรือไม่; หากไม่ใช่โปรแกรมจะแจ้ง "Invalid choice." แล้วกลับไปแสดงเมนูใหม่ ถ้าข้อมูลเป็นตัวเลข จะเปลี่ยนเป็นชนิดจำนวนเต็มด้วย `int(s)` แล้วเก็บในตัวแปร `choice` ต่อมาโค้ดใช้ชุดคำสั่ง `if/elif` เพื่อตรวจค่า `choice` และเรียกฟังก์ชันที่สอดคล้องกับเมนู เช่น `add_book()` สำหรับตัวเลือก 1, `add_member()` สำหรับ 2, `add_loan()` สำหรับ 3 เป็นต้น ตัวเลือก 4 เรียก `return_loan()` เพื่อคืนหนังสือ, 5 เรียก `update_book()` เพื่อแก้ไขข้อมูลหนังสือ, 6 เรียก `delete_book()` เพื่อ (น่าจะ)ลบหนังสือ, และ 7 เรียก `generate_report()` เพื่อสร้างรายงานเป็นไฟล์ .txt ถ้าผู้ใช้กรอก 0 โปรแกรมจะแสดง "Exiting..." แล้วใช้ `break` เพื่อออกจากลูปและจบการทำงานของเมนู. ถ้าค่าที่กรอกเป็นตัวเลขแต่ไม่ตรงกับตัวเลือกใด ๆ โค้ดจะเข้าสู่บล็อก `else` และพิมพ์ "Invalid option" ฟังก์ชันที่ถูกเรียกจากเมนู (`add_book`, `add_member`, `add_loan`, ฯลฯ) ต้องถูกนิยามไว้ที่อื่นในโปรแกรม มิฉะนั้นจะเกิดข้อผิดพลาดเมื่อเรียกใช้งาน โดยรวมโครงสร้างนี้เป็นเมนูแบบ `synchronous` ที่รับคำสั่งจากผู้ใช้ ตรวจสอบความถูกต้องขั้นต้นของอินพุต และเรียกฟังก์ชันที่ทำงานจริงตามตัวเลือก

```
def main_menu():
    while True:
        print(textwrap.dedent('''
            ===== Library Manager =====
            1) Book (Add/Update/Delete)
            2) Member (Add/Update/Delete)
            3) Borrow Book
            4) Return Book
            5) View
            6) Generate Report (.txt)
            0) Exit
        '''))
        s = input("Select: ").strip()

        if not s.isdigit():
            print("Invalid choice.")
            continue

        choice = int(s)
        if choice == 1:
            # Book submenu
            print(textwrap.dedent('''
                --- Book Menu ---
                1) Add Book
                2) Update Book
                3) Delete Book
                0) Back
            '''))
            sub = input("Select: ").strip()
            if sub == "1":
                add_book()
            elif sub == "2":
                update_book()
            elif sub == "3":
                delete_book()
            elif sub == "0":
                continue
            else:
                print("Invalid option.")
```

รูปภาพ 4-1 main_menu()

```

elif choice == 2:
    # Member submenu
    print(textwrap.dedent('''
        --- Member Menu ---
        1) Add Member
        2) Update Member
        3) Delete Member
        0) Back
    '''))
    sub = input("Select: ").strip()
    if sub == "1":
        add_member()
    elif sub == "2":
        update_member()
    elif sub == "3":
        delete_member()
    elif sub == "0":
        continue
    else:
        print("Invalid option.")

elif choice == 3:
    add_loan()
elif choice == 4:
    return_loan()
elif choice == 5:
    view()
elif choice == 6:
    generate_report()
elif choice == 0:
    print("Exiting...")
    break
else:
    print("Invalid option.")

```

รูปภาพ 4-2 main_menu() ต่อ

4.2.3 add_book()

ฟังก์ชัน add_book() ใช้สำหรับเพิ่มหนังสือใหม่ลงในระบบ โดยเริ่มจากแสดงข้อความ "-- Add Book --" เพื่อบอกผู้ใช้งานว่ากำลังอยู่ในโหมดเพิ่มหนังสือ จากนั้นระบบจะให้ผู้ใช้งานกรอกข้อมูลของหนังสือ ได้แก่ ชื่อหนังสือ (Title), ชื่อผู้แต่ง (Author), ปีพิมพ์ (Year) และจำนวนเล่มทั้งหมด (Total copies) หลังจากนั้นจะสร้างหมายเลขประจำตัวหนังสือ (ID) ใหม่โดยอัตโนมัติ ระบบจะแสดงข้อมูลหนังสือทั้งหมดให้ผู้ใช้งานตรวจสอบอีกครั้งก่อนบันทึก หากผู้ใช้งานยืนยันด้วยการพิมพ์ y หรือ yes ข้อมูลจะถูกจัดเก็บลงในไฟล์ โดยใช้การเข้ารหัส (pack) ตามโครงสร้างที่กำหนดและเพิ่มเวลาในการบันทึก

(timestamp) สุดท้ายจะแสดงข้อความยืนยันว่าหนังสือถูกเพิ่มเรียบร้อยแล้ว หากผู้ใช้ไม่ยืนยัน การเพิ่มหนังสือจะถูกยกเลิกทันที ฟังก์ชัน `update_book()` ใช้สำหรับแก้ไขข้อมูลหนังสือที่มีอยู่แล้ว โดยเริ่มจากแสดงรายการหนังสือทั้งหมดในรูปแบบตารางเพื่อให้ผู้ใช้เลือก ID ของหนังสือที่ต้องการแก้ไข หลังจากกรอก ID ระบบจะตรวจสอบว่าหนังสือมีอยู่จริงหรือไม่ หากไม่พบจะแจ้งข้อความว่า "Book not found" แต่หากพบ ระบบจะให้ผู้ใช้กรอกข้อมูลใหม่ โดยสามารถเว้นว่างไว้เพื่อคงค่าปัจจุบันของหนังสือแต่ละฟิลด์ หลังจากกรอกข้อมูลใหม่ ระบบจะคำนวณจำนวนเล่มที่ว่างให้ยืม (Available) ใหม่โดยอัตโนมัติ และแสดงข้อมูลหนังสือที่อัปเดตให้ผู้ใช้ตรวจสอบอีกครั้ง หากผู้ใช้ยืนยัน ระบบจะบันทึกการเปลี่ยนแปลงลงในไฟล์ด้วยการเข้ารหัสตามโครงสร้างเดิม และแสดงข้อความยืนยันว่า "Book updated successfully" หากไม่ยืนยัน การแก้ไขจะถูกยกเลิก

```
def add_book():
    print("-- Add Book --")
    title = safe_input("Title: ")
    author = safe_input("Author: ")
    year = safe_input("Year (e.g., 2024): ", int)
    total = safe_input("Total copies: ", int)
    bid = get_next_id(BOOKS_FILE, BOOK_STRUCT)

    print("\nPlease confirm the book information:")
    print(f" ID      : {bid}")
    print(f" Title   : {title}")
    print(f" Author  : {author}")
    print(f" Year    : {year}")
    print(f" Total   : {total}")
    print(f" Avail.  : {total}")

    confirm = input("Save this book? (y/n): ").strip().lower()
    if confirm not in ["y", "yes"]:
        print("Add book canceled.")
        return

    packed = BOOK_STRUCT.pack(
        bid,
        pack_fixed_str(title, TITLE_LEN),
        pack_fixed_str(author, AUTHOR_LEN),
        int(year),
        int(total),
        int(total),
        1,
        now_ts()
    )
    append_record(BOOKS_FILE, BOOK_STRUCT, packed)
    print(f"Added Book id={bid}")
```

รูปภาพ 4-3 `add_book()`

4.2.2 `add_member()`

ฟังก์ชัน `add_member()` ใช้สำหรับเพิ่มสมาชิกใหม่ลงในระบบ โดยเริ่มจากแสดงข้อความ "-- Add Member --" เพื่อบอกผู้ใช้ว่ากำลังอยู่ในโหมดเพิ่มสมาชิก จากนั้นระบบจะให้ผู้ใช้กรอกข้อมูลสมาชิก

ได้แก่ ชื่อเต็ม (Full Name), เบอร์โทรศัพท์ (Phone) และที่อยู่ (Address) หลังจากนั้นจะสร้างหมายเลขประจำตัวสมาชิก (ID) ใหม่โดยอัตโนมัติ ระบบจะแสดงข้อมูลสมาชิกทั้งหมดให้ผู้ใช้ตรวจสอบอีกครั้งก่อนบันทึก หากผู้ใช้ยืนยันด้วยการพิมพ์ y หรือ yes ข้อมูลจะถูกจัดเก็บลงในไฟล์โดยใช้การเข้ารหัส (pack) ตามโครงสร้างสมาชิกและเพิ่มค่า active เป็น 1 พร้อมบันทึกเวลาในการเพิ่มสมาชิก (timestamp) สุดท้ายจะแสดงข้อความยืนยันว่ามีการเพิ่มสมาชิกเรียบร้อยแล้ว หากผู้ใช้ไม่ยืนยัน การเพิ่มสมาชิกจะถูกยกเลิกทันทีฟังก์ชัน list_members() ใช้สำหรับดึงรายชื่อสมาชิกทั้งหมดจากไฟล์ โดยเริ่มจากอ่านข้อมูลทุกแถวจากไฟล์สมาชิกและถอดรหัสแต่ละฟิลด์ จากนั้นตรวจสอบสถานะสมาชิก หากไม่ได้ระบุให้แสดงสมาชิกที่ไม่ใช้งาน (show_inactive=False) ระบบจะข้ามสมาชิกที่ถูกลบ (active == 0) และจัดเก็บสมาชิกที่ใช้งานอยู่ในรูปแบบรายการของดิกชันนารี (dictionary) ซึ่งประกอบด้วยหมายเลขแถว, ID, ชื่อ, เบอร์โทรศัพท์, ที่อยู่, สถานะใช้งาน และเวลาแก้ไขล่าสุด (last_modified) สุดท้ายฟังก์ชันจะคืนค่ารายชื่อสมาชิกทั้งหมดตามเงื่อนไขที่กำหนด

```
def add_member():
    print("-- Add Member --")
    name = safe_input("Full Name: ")
    phone = safe_input("Phone: ")
    addr = safe_input("Address: ")
    mid = get_next_id(MEMBERS_FILE, MEMBER_STRUCT)

    print("\nPlease confirm the information:")
    print(f" ID      : {mid}")
    print(f" Name    : {name}")
    print(f" Phone   : {phone}")
    print(f" Addr    : {addr}")

    confirm = input("Save this member? (y/n): ").strip().lower()
    if confirm not in ["y", "yes"]:
        print("Add member canceled.")
        return

    packed = MEMBER_STRUCT.pack(
        mid,
        pack_fixed_str(name, NAME_LEN),
        pack_fixed_str(phone, PHONE_LEN),
        pack_fixed_str(addr, ADDR_LEN),
        1, # active
        now_ts()
    )
    append_record(MEMBERS_FILE, MEMBER_STRUCT, packed)
    print(f"Added Member id={mid}")
```

รูปภาพ 4-4 add_member()

4.2.4 add_loan()

ฟังก์ชัน add_loan() ทำหน้าที่เป็นเมนูสำหรับให้ผู้ใช้ยืมหนังสือ โดยเริ่มจากพิมพ์หัวข้อว่าเป็นขั้นตอนการยืมหนังสือเพื่อให้ผู้ใช้ทราบ จากนั้นเรียก list_books() มาดูรายการหนังสือ ถ้าไม่มี

หนังสือในระบบฟังก์ชันจะพิมพ์เดือนแล้วคืนค่าออกทันที ถ้ามีหนังสือจะแสดงตารางสรุปรายการหนังสือ (ID, Title, Author, Year, Available) โดยวนแสดงแต่ละเล่มให้ผู้เลือกใช้ ผู้ใช้กรอก Book ID ผ่าน `safe_input(..., int)` เพื่อเลือกเล่มที่ต้องการ แล้วใช้ `find_book_by_id(bid)` หาเรคอร์ดหนังสือจริง; หากไม่พบจะพิมพ์ข้อความเตือนแล้วจบการทำงานถ้าพบหนังสือแล้วฟังก์ชันจะตรวจสอบว่า `book["available"]` มากกว่า 0 หากไม่เหลือจะบอกว่า “หนังสือไม่เหลือให้ยืม” แล้วคืนค่า ต่อไปจะเรียก `list_members()` เพื่อแสดงรายชื่อสมาชิกทั้งหมด ถ้าไม่มีสมาชิกก็จะแจ้งและคืนค่า ถ้ามีจะแสดงตารางสมาชิกให้ผู้เลือกใช้ Member ID. ผู้ใช้กรอก Member ID ทาง `safe_input(..., int)` แล้วใช้ `find_member_by_id(mid)` ตรวจสอบความถูกต้องของสมาชิก ถ้าไม่พบก็แจ้งแล้วออก เมื่อทั้งหนังสือและสมาชิกถูกตรวจสอบเรียบร้อยแล้ว ฟังก์ชันจะสร้าง Loan ID ใหม่ด้วย `get_next_id()` แล้วใช้ `LOAN_STRUCT.pack(...)` บันทึกข้อมูลการยืม (LoanID, BookID, MemberID, borrow_timestamp, return_timestamp=0, status=1, created_at) และเขียนเรคอร์ดลงไฟล์ด้วย `append_record()` หลังจากบันทึกการยืมแล้วโค้ดจะอัปเดตจำนวนหนังสือที่มีอยู่จริงโดยอ่านเรคอร์ดหนังสือทั้งหมด (`read_all_records()`), หาตำแหน่งของเล่มที่ยืม แล้วเขียนทับเรคอร์ดนั้นโดยลด `available` ลงหนึ่งและอัปเดต `timestamp` ด้วย `write_record_at()` สุดท้ายฟังก์ชันจะแสดงข้อความยืนยันว่า การยืมสำเร็จโดยระบุชื่อสมาชิก ชื่อหนังสือ และ LoanID เพื่อให้ผู้ใช้ทราบว่าธุรกรรมเสร็จสมบูรณ์

```
def add_loan():
    print("-- Borrow Book --")

    books = list_books()
    if not books:
        print("ไม่มีหนังสือในระบบ")
        return
    print("\nAvailable Books:")
    print("ID | Title | Author | Year | Available")
    for b in books:
        print(f"{b['id']:<2} | {b['title'][:28]:<28} | {b['author'][:12]} | {b['year'][:4]} | {b['available']}]")

    bid = safe_input("\nBook ID ที่ต้องการยืม: ", int)
    _, book = find_book_by_id(bid)
    if not book:
        print("ไม่พบหนังสือ")
        return
    if book["available"] <= 0:
        print("หนังสือไม่เหลือให้ยืม")
        return

    members = list_members()
    if not members:
        print("ไม่มีสมาชิกในระบบ")
        return
    print("\nMembers:")
    print("ID | Name | Phone")
    for m in members:
        print(f"{m['id']:<2} | {m['name'][:20]} | {m['phone']}")

    mid = safe_input("\nMember ID ผู้ยืม: ", int)
    _, mem = find_member_by_id(mid)
    if not mem:
        print("ไม่พบสมาชิก")
        return
```

รูปภาพ 4-5 add_loan()

```

members = list_members()
if not members:
    print("ไม่มีสมาชิกในระบบ")
    return
print("\nMembers:")
print("ID | Name | Phone")
for m in members:
    print(f"{m['id']:<2} | {m['name']:<20} | {m['phone']}")

mid = safe_input("\nMember ID ยืม: ", int)
_, mem = find_member_by_id(mid)
if not mem:
    print("ไม่พบสมาชิก")
    return

print("\nPlease confirm borrow information:")
print(f" LoanID : (auto) next ID")
print(f" Book : {book['title']} (ID={bid})")
print(f" Member : {mem['name']} (ID={mid})")

while True:
    confirm = input("Confirm borrow? (y/n): ").strip().lower()
    if confirm in ["y", "yes"]:
        break
    elif confirm in ["n", "no"]:
        print("Borrow canceled.")
        return
    else:
        print("Please enter 'y' or 'n'.")

lid = get_next_id(LOANS_FILE, LOAN_STRUCT)
packed = LOAN_STRUCT.pack(lid, bid, mid, now_ts(), 0, 1, now_ts())
append_record(LOANS_FILE, LOAN_STRUCT, packed)

recs = read_all_records(BOOKS_FILE, BOOK_STRUCT)

```

รูปภาพ 4-6 add_loan() ต่อ 1

```

recs = read_all_records(BOOKS_FILE, BOOK_STRUCT)
for idx, v in recs:
    if v[0] == bid:
        packed_b = BOOK_STRUCT.pack(v[0], v[1], v[2], v[3], v[4], v[5] - 1, v[6], now_ts())
        write_record_at(BOOKS_FILE, BOOK_STRUCT, idx, packed_b)
        break

print(f"\nBorrow success: {mem['name']} ยืม {book['title']} (LoanID={lid})")

```

รูปภาพ 4-7 add_loan() ต่อ 2

4.2.5 return_loan()

ฟังก์ชัน return_loan() ทำหน้าที่ในการจัดการคืนหนังสือที่สมาชิกเคยยืมไว้ โดยเริ่มแรกจะพิมพ์หัวข้อ “คืนหนังสือ” ออกมาเพื่อบอกผู้ใช้ จากนั้นโปรแกรมจะค้นหารายการยืมทั้งหมดที่ยังไม่ถูกคืน ซึ่งจะตรวจสอบจากฟิลด์ return_date ที่ต้องมีค่าเท่ากับศูนย์ ถ้าไม่มีรายการใดที่ยืมค้างอยู่ โปรแกรมจะแจ้งว่าไม่มีรายการที่ต้องคืนแล้วจบการทำงาน แต่ถ้ามีรายการยืมค้างอยู่ โปรแกรมจะ

ดึงข้อมูลหนังสือและข้อมูลสมาชิกที่เกี่ยวข้องทั้งหมดมาเก็บไว้ในรูปแบบพจนานุกรมเพื่อง่ายต่อการค้นหา ถัดมา โปรแกรมจะแสดงตารางที่สรุปรายการยืมที่ยังไม่ถูกคืน โดยแสดง LoanID, BookID, ชื่อหนังสือ, MemberID, ชื่อสมาชิก และวันที่ยืม เพื่อให้ผู้ใช้ดูได้ชัดเจนว่าใครยืมหนังสืออะไรอยู่ จากนั้นผู้ใช้จะต้องป้อน LoanID ของรายการที่ต้องการคืน เมื่อได้ LoanID แล้ว โปรแกรมจะไปค้นหาในไฟล์เก็บข้อมูลการยืม (loans.bin) เพื่อตรวจสอบว่ามีรายการนี้จริงหรือไม่ ถ้าไม่พบจะแจ้งว่าไม่พบรายการ แต่ถ้าพบก็จะตรวจสอบต่อไปว่ารายการนั้นได้ถูกคืนไปแล้วหรือยัง ถ้าถูกคืนแล้วจะแจ้งเตือน แต่ถ้ายังไม่ได้คืน โปรแกรมจะบันทึกเวลาคืนปัจจุบันลงในฟิลด์ return_date ของระเบียนนั้น แล้วเขียนทับกลับไปไฟล์ หลังจากอัปเดตข้อมูลการคืนเรียบร้อยแล้ว โปรแกรมยังไปอัปเดตไฟล์หนังสือ (books.bin) โดยเพิ่มจำนวนเล่มที่ยังว่าง (available) ขึ้น 1 เพื่อสะท้อนว่ามีหนังสือกลับเข้ามาในระบบแล้ว สุดท้าย โปรแกรมจะแสดงข้อความยืนยันว่าได้คืนหนังสือเรียบร้อยแล้ว พร้อมบอกชื่อสมาชิกและชื่อหนังสือที่ถูกคืน

```
def return_loan():
    print("-- Return Loan (คืนหนังสือ) --")

    # ดึงรายการยืมที่ยังไม่ถูกคืน (return_date=0)
    loans = [l for l in list_loans() if l["return_date"] == 0]
    if not loans:
        print("ไม่มีรายการที่ต้องคืน")
        return

    books = {b["id"]: b for b in list_books()}
    members = {m["id"]: m for m in list_members()}

    # แสดงตารางรายการที่ยังไม่ถูกคืน
    print("\nรายการที่ยังไม่ถูกคืน:")
    print("LoanID | BookID | Title | MemberID | Member Name | Borrow Date")
    print("-----")
    for l in loans:
        b = books.get(l["book_id"], {})
        m = members.get(l["member_id"], {})
        print(f"{l['id']:<6} | {l['book_id']:<6} | {b.get('title', '-')[:20]:<20} | "
              f"{l['member_id']:<8} | {m.get('name', '-')[:18]} | {fmt_ts(l['borrow_date'])}")
```

รูปภาพ 4-8 return_loan()

```

lid = safe_input("\nกรอก LoanID ที่ต้องการคืน: ", int)

recs = read_all_records(LOANS_FILE, LOAN_STRUCT)
target_idx, target = None, None
for idx, vals in recs:
    if vals[0] == lid:
        target_idx, target = idx, vals
        break

if not target:
    print("ไม่พบรายการยืมนี้")
    return

(rid, book_id, member_id, borrow_ts, return_ts, active, last_mod) = target
if return_ts != 0:
    print("รายการนี้คืนไปแล้ว")
    return

```

รูปภาพ 4-9 return_loan() ต่อ

4.2.6 generate_report()

ฟังก์ชัน generate_report() ใช้สำหรับสร้างรายงานสรุปข้อมูลทั้งหมดของระบบยืมหนังสือ (Library Borrow System) โดยจะดึงข้อมูลจากไฟล์หนังสือ สมาชิก และประวัติการยืม แล้วจัดรูปแบบเป็นรายงานในลักษณะตารางและสรุปข้อมูลสำคัญ จากนั้นแสดงผลทางหน้าจอและบันทึกลงในไฟล์รายงาน เมื่อเริ่มทำงาน ฟังก์ชันจะอ่านข้อมูลทั้งหมดจากสามส่วนหลัก ได้แก่ รายชื่อหนังสือ (list_books), รายชื่อสมาชิก (list_members), และประวัติการยืม (list_loans) โดยกำหนดให้แสดงข้อมูลแม้จะเป็นรายการที่ถูกลบ (ใช้ show_inactive=True) จากนั้นจะสร้างโครงสร้างข้อมูลแบบพจนานุกรม (book_map และ member_map) เพื่อให้ง่ายต่อการค้นหาข้อมูลหนังสือหรือสมาชิกจาก ID ต่อมาจะเริ่มต้นสร้างข้อความรายงาน (lines) โดยเพิ่มหัวข้อมรายงาน เช่น ชื่อระบบ ("Library Borrow System - Report"), วันที่และเวลาที่สร้างรายงาน, เวอร์ชันของโปรแกรม และรูปแบบการเข้ารหัสที่ใช้ จากนั้นเข้าสู่ส่วนของ Borrow History ซึ่งเป็นตารางแสดงประวัติการยืมหนังสือทั้งหมด ในส่วนนี้ โปรแกรมจะตรวจสอบว่ามีข้อมูลการยืมหรือไม่ หากมี จะทำการจัดกลุ่มข้อมูลตามรหัสสมาชิก (member_id) เพื่อรวมรายการยืมหลายเล่มของสมาชิกแต่ละคนเข้าด้วยกัน โดยข้อมูลแต่ละรายการจะประกอบด้วยชื่อสมาชิก, เบอร์โทรศัพท์, ชื่อหนังสือ, วันที่ยืม, วันที่คืน และสถานะการยืม (เช่น "Borrowed" หรือ "Returned") ถ้าสมาชิกยืมหลายเล่ม โปรแกรมจะรวมชื่อหนังสือทั้งหมดไว้ในบรรทัดเดียว เพื่อให้รายงานอ่านง่ายและไม่ซ้ำซ้อน แต่ถ้าไม่มีข้อมูลการยืม ระบบจะแสดงบรรทัดว่างแทน หลังจากนั้นจะเข้าสู่ส่วน Summary ซึ่งเป็นการสรุปภาพรวมของหนังสือในระบบ โดยแสดงจำนวนหนังสือทั้งหมด, จำนวนหนังสือที่ยังใช้งานอยู่ (Active), หนังสือที่ถูกลบ (Deleted), จำนวนหนังสือที่กำลังถูกยืมอยู่ (Borrowed Now) และจำนวนหนังสือที่ยังว่างให้ยืม (Available Now) ต่อมาคือส่วนของ Borrow Statistics ซึ่งใช้

สำหรับวิเคราะห์ข้อมูลการยืม โดยโปรแกรมจะนับจำนวนครั้งที่หนังสือแต่ละเล่มถูกยืม (โดยใช้คลาส Counter) จากนั้นหาหนังสือที่ถูกยืมบ่อยที่สุด และแสดงชื่อหนังสือนั้นพร้อมจำนวนครั้งที่ถูกยืม รวมถึงจำนวนหนังสือที่กำลังถูกยืมอยู่ในขณะนั้น และจำนวนสมาชิกที่ยังคงใช้งานอยู่ในระบบ เมื่อรวมข้อความทั้งหมดเสร็จ ฟังก์ชันจะนำข้อมูลในรายการ (lines) มารวมกันเป็นข้อความเดียว (report_text) จากนั้นพิมพ์รายงานออกทางหน้าจอเพื่อให้ผู้ใช้ดูได้ทันที และบันทึกข้อมูลรายงานทั้งหมดลงในไฟล์ที่กำหนด (path หรือ REPORT_FILE) สุดท้ายจะแสดงข้อความแจ้งว่ารายงานถูกบันทึกเรียบร้อยแล้ว พร้อมระบุเส้นทางไฟล์ที่จัดเก็บ

```
def generate_report(path=REPORT_FILE):
    books = list_books(show_inactive=True)
    members = list_members(show_inactive=True)
    loans = list_loans(show_inactive=True)

    book_map = {b["id"]: b for b in books}
    member_map = {m["id"]: m for m in members}

    lines = []
    lines.append("Library Borrow System - Report")
    lines.append(f"Generated At : {datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')}")
    lines.append("App Version : 3.0")
    lines.append("Encoding : UTF-8\n")

    # ----- BORROW HISTORY -----
    lines.append("Borrow History")
    lines.append("-----")
    lines.append("MemberID | Member Name | Phone | Title | Loan Date | Return Date | Status |")
    lines.append("-----")

    if loans:
        grouped = {}
        for l in loans:
            mid = l["member_id"]
            m = member_map.get(mid, {})
            b = book_map.get(l["book_id"], {})
            loan_date = fmt_ts[l["borrow_date"]]
            return_date = fmt_ts[l["return_date"]]
            status = "Returned" if l["return_date"] else "Borrowed"

            if mid not in grouped:
                grouped[mid] = {
                    "member": m,
                    "titles": [b.get("title", "-")],
                    "loan_dates": [loan_date],
                    "return_dates": [return_date],
                    "status": [status]
                }
            else:
                grouped[mid]["titles"].append(b.get("title", "-"))
                grouped[mid]["loan_dates"].append(loan_date)
                grouped[mid]["return_dates"].append(return_date)
                grouped[mid]["status"].append(status)

        for mid, data in grouped.items():
            m = data["member"]
            titles = "; ".join(data["titles"][:27])
            loan_d = data["loan_dates"][0] if data["loan_dates"] else "-"
            ret_d = data["return_dates"][0] if data["return_dates"] else "-"
            status = data["status"][0] if data["status"] else "-"

            lines.append(
                f"|{m.get('id',''):<9}"
                f"|{m.get('name','') or '-':<30}"
                f"|{m.get('phone','') or '-':<12}"
                f"|{titles:<27}"
                f"|{loan_d:<12}"
                f"|{ret_d:<12}"
                f"|{status:<10}"
            )
        else:
            lines.append(" | - | - | - | - | - | ")

    lines.append(
        "-----"
        # ----- SUMMARY -----
    )
    active_books = [b for b in books if b["active"] == 1]
    deleted_books = [b for b in books if b["active"] == 0]
    borrowed_now = len([l for l in loans if l["return_date"] == 0])
    available_now = sum([b["available"] for b in active_books])

    lines.append("Summary (Active Books Only)")
    lines.append(f"Total Books : {len(books)}")
    lines.append(f"Active Books : {len(active_books)}")
```

รูปภาพ 4-10 generate_report()

```
else:
    grouped[mid]["titles"].append(b.get("title", "-"))
    grouped[mid]["loan_dates"].append(loan_date)
    grouped[mid]["return_dates"].append(return_date)
    grouped[mid]["status"].append(status)

for mid, data in grouped.items():
    m = data["member"]
    titles = "; ".join(data["titles"][:27])
    loan_d = data["loan_dates"][0] if data["loan_dates"] else "-"
    ret_d = data["return_dates"][0] if data["return_dates"] else "-"
    status = data["status"][0] if data["status"] else "-"

    lines.append(
        f"|{m.get('id',''):<9}"
        f"|{m.get('name','') or '-':<30}"
        f"|{m.get('phone','') or '-':<12}"
        f"|{titles:<27}"
        f"|{loan_d:<12}"
        f"|{ret_d:<12}"
        f"|{status:<10}"
    )
else:
    lines.append(" | - | - | - | - | - | ")

lines.append(
    "-----"
    # ----- SUMMARY -----
)
active_books = [b for b in books if b["active"] == 1]
deleted_books = [b for b in books if b["active"] == 0]
borrowed_now = len([l for l in loans if l["return_date"] == 0])
available_now = sum([b["available"] for b in active_books])

lines.append("Summary (Active Books Only)")
lines.append(f"Total Books : {len(books)}")
lines.append(f"Active Books : {len(active_books)}")
```

รูปภาพ 4-11 generate_report() ต่อ

```

active_books = [b for b in books if b['active'] == 1]
deleted_books = [b for b in books if b['active'] == 0]
borrowed_now = len([l for l in loans if l['return_date'] == 0])
available_now = sum([b['available'] for b in active_books])

lines.append("Summary (Active Books Only)")
lines.append(f"- Total Books      : {len(books)}")
lines.append(f"- Active Books       : {len(active_books)}")
lines.append(f"- Deleted Books      : {len(deleted_books)}")
lines.append(f"- Borrowed Now       : {borrowed_now}")
lines.append(f"- Available Now      : {available_now}\n")

# ----- BORROW STATISTICS -----
from collections import Counter
borrow_count = Counter([l['book_id'] for l in loans])
most_borrowed = borrow_count.most_common(1)

lines.append("Borrow Statistics (Active only)")
if most_borrowed:
    book_id, times = most_borrowed[0]
    lines.append(f"- Most Borrowed Book : {book_map.get(book_id, {}).get('title', '-')} ({times} times)")
else:
    lines.append("- Most Borrowed Book : None")

lines.append(f"- Currently Borrowed : {borrowed_now}")
lines.append(f"- Active Members      : {len([m for m in members if m['active'] == 1])}")

# รวมเป็น text
report_text = "\n".join(lines)

print(report_text)

with open(path, "w", encoding="utf-8") as f:
    f.write(report_text)

```

รูปภาพ 4-12 generate_report() ต่อ

4.2.7 update_book()

เมื่อผู้ใช้เลือกเมนู **Update Book** โปรแกรมจะเริ่มต้นด้วยการแสดงข้อความหัวข้อว่า “-- Update Book --” เพื่อระบุว่ากำลังเข้าสู่โหมดแก้ไขข้อมูลหนังสือ จากนั้นจะทำการดึงข้อมูลหนังสือทั้งหมดจากระบบด้วยฟังก์ชัน `list_books()` และแสดงผลในรูปแบบตารางที่มีคอลัมน์แสดง ID, Title, Author, Year, Total และ Available เพื่อให้ผู้ใช้เห็นรายการหนังสือทั้งหมดก่อนเลือกที่จะอัปเดตเล่มใด หลังจากแสดงตารางแล้ว โปรแกรมจะให้ผู้กรอก **Book ID** ที่ต้องการอัปเดต โดยใช้ฟังก์ชัน `safe_input()` เพื่อรับค่าเป็นตัวเลข หากไม่พบหนังสือที่มีรหัสดังกล่าว โปรแกรมจะแจ้งว่า “Book not found” และหยุดการทำงานทันที เมื่อพบข้อมูลหนังสือที่ต้องการแก้ไขแล้ว โปรแกรมจะให้ผู้กรอกข้อมูลใหม่ในแต่ละช่อง ได้แก่ Title, Author, Year และ Total copies โดยในแต่ละช่องจะมีข้อความกำกับกับค่าปัจจุบันไว้ในวงเล็บ เช่น “Title [Old Title]:” เพื่อให้ผู้ใช้ทราบค่าดั้งเดิม และสามารถปล่อยว่างได้หากต้องการเก็บค่าเดิมไว้เหมือนเดิม หลังจากรับข้อมูลใหม่แล้ว โปรแกรมจะนำค่าที่ผู้กรอกมาสร้างเป็นค่าที่อัปเดตใหม่ หากช่องใดไม่ได้กรอกจะใช้ค่าปัจจุบันแทน แล้วคำนวณค่าความแตกต่างของจำนวนเล่มทั้งหมด (diff) เพื่อปรับปรุงจำนวนเล่มที่ “Available” โดยเพิ่มหรือลดตามความต่างของจำนวนรวม (Total) เพื่อให้จำนวนหนังสือที่ยืมได้สะท้อนตามความจริง เมื่อได้ข้อมูลที่อัปเดตแล้ว โปรแกรมจะแสดงรายละเอียดหนังสือใหม่ทั้งหมดอีกครั้ง เช่น ID, Title, Author, Year, Total, Available และ Status เพื่อให้ผู้ใช้ตรวจสอบความถูกต้องก่อนบันทึก พร้อมให้ยืนยันด้วยการพิมพ์ y หรือ yes เพื่อยืนยันการอัปเดต หรือพิมพ์อย่างอื่นเพื่อยกเลิกการทำการรายการ หากผู้ใช้เลือกยืนยัน (y หรือ yes) โปรแกรมจะจัดเก็บข้อมูลที่อัปเดตโดยใช้

BOOK_STRUCT.pack() เพื่อบันทึกข้อมูลใหม่กลับเข้าไฟล์จริงผ่านฟังก์ชัน write_record_at() ซึ่งจะอัปเดตเฉพาะแถวของหนังสือที่แก้ไขเท่านั้น จากนั้นโปรแกรมจะแสดงข้อความว่า “Book updated successfully.” เพื่อแจ้งว่าการอัปเดตสำเร็จเรียบร้อย

```
def delete_book():
    print("-- Delete Book --")

    books = list_books(show_inactive=True)
    if not books:
        print("No books in system")
        return

    print("+-----+-----+-----+-----+-----+-----+")
    print("| ID | Title | Author | Year | Total | Available|")
    print("+-----+-----+-----+-----+-----+-----+")
    for b in books:
        status = "Active" if b["active"] else "Deleted"
        print(f"|{b['id']:<3} | {b['title'][:25]:<25} | {b['author'][:16]:<16} | "
              f"{b['year']:<4} | {b['total']:<5} | {b['available']:<8} | {status}")
    print("+-----+-----+-----+-----+-----+-----+")

    bid = safe_input("Enter Book ID to delete: ", int)
    idx, book = find_book_by_id(bid)
    if not book:
        print("Book not found")
        return

    if book["active"] == 0:
        print("Book already deleted")
        return

    active_loans = [l for l in list_loans() if l["book_id"] == bid and l["return_date"] == 0]
    if active_loans:
        print("Cannot delete book: it is currently borrowed by someone.")
        return

    confirm = input(f"Are you sure you want to delete '{book['title']}' (y/n)? ").strip().lower()
    if confirm not in ["y", "yes"]:
        print("Delete canceled.")
        return

    packed = BOOK_STRUCT.pack(
        book["id"],
        pack_fixed_str(book["title"], TITLE_LEN),
```

รูปภาพ 4-13 update_book()

```
        pack_fixed_str(book["title"], TITLE_LEN),
        pack_fixed_str(book["author"], AUTHOR_LEN),
        book["year"],
        book["total"],
        book["available"],
        0, # inactive
        now_ts()
    )
    write_record_at(BOOKS_FILE, BOOK_STRUCT, idx, packed)
    print(f"Book '{book['title']}' soft-deleted successfully.")
```

รูปภาพ 4-14 update_book() ต่อ

4.2.8 delete_book()

ฟังก์ชัน `delete_book()` ทำงานดังนี้: ฟังก์ชันจะแสดงข้อความว่าเป็นเมนู "Delete Book" และดึงรายการหนังสือทั้งหมดด้วย `list_books()` เพื่อให้ผู้ใช้เห็นข้อมูลปัจจุบัน จากนั้นจะแสดงรายละเอียดหนังสือทั้งหมดให้ผู้ใช้เลือก โดยให้ผู้ใช้กรอก **Book ID** ของหนังสือที่ต้องการลบ หากไม่พบหนังสือที่กรอก ระบบจะแจ้งว่าไม่พบหนังสือและออกจากฟังก์ชัน ถ้าพบหนังสือแต่สถานะ `active` เป็น 0 แสดงว่าลบไปแล้ว ระบบจะแจ้งว่า "Book already deleted" และออกจากฟังก์ชัน ถ้าหนังสือยังใช้งานอยู่ ฟังก์ชันจะทำการแฟกข้อมูลเดิมของหนังสือ แต่เปลี่ยนค่า `active` เป็น 0 เพื่อทำ **soft delete** และเขียนข้อมูลกลับไปยังไฟล์ไบนารี `books.bin` ที่ตำแหน่งเดิม สุดท้ายจะแจ้งผู้ใช้ว่า "Book deleted (soft delete)."

```
def delete_book():
    print("-- Delete Book --")

    books = list_books(show_inactive=True)
    if not books:
        print("No books in system")
        return

    print("+---+-----+-----+-----+-----+-----+")
    print("| ID | Title | Author | Year | Total | Available|")
    print("+---+-----+-----+-----+-----+-----+")
    for b in books:
        status = "Active" if b["active"] else "Deleted"
        print(f"|{b['id']:<3}| {b['title'][:25]:<25}| {b['author'][:16]:<16}| "
              f"|{b['year']:<4}| {b['total']:<5}| {b['available']:<8}| {status}|")
    print("+---+-----+-----+-----+-----+-----+")

    bid = safe_input("Enter Book ID to delete: ", int)
    idx, book = find_book_by_id(bid)
    if not book:
        print("Book not found")
        return

    if book["active"] == 0:
        print("Book already deleted")
        return

    active_loans = [l for l in list_loans() if l["book_id"] == bid and l["return_date"] == 0]
    if active_loans:
        print("Cannot delete book: it is currently borrowed by someone.")
        return

    confirm = input(f"Are you sure you want to delete '{book['title']}' (y/n)? ").strip().lower()
    if confirm not in ["y", "yes"]:
        print("Delete canceled.")
        return
```

รูปภาพ 4-15 delete_book()

```

packed = BOOK_STRUCT.pack(
    book["id"],
    pack_fixed_str(book["title"], TITLE_LEN),
    pack_fixed_str(book["author"], AUTHOR_LEN),
    book["year"],
    book["total"],
    book["available"],
    0, # inactive
    now_ts()
)
write_record_at(BOOKS_FILE, BOOK_STRUCT, idx, packed)
print(f"Book '{book['title']}' soft-deleted successfully.")

```

รูปภาพ 4-16 delete_book() ต่อ

4.2.9 Update_member()

ฟังก์ชัน update_member() ใช้สำหรับแก้ไขข้อมูลของสมาชิกที่มีอยู่ในระบบ โดยเริ่มจากแสดงข้อความ “-- Update Member --” เพื่อแจ้งผู้ใช้งานที่กำลังอยู่ในโหมดแก้ไขข้อมูลสมาชิก จากนั้นโปรแกรมจะดึงรายชื่อสมาชิกทั้งหมดจากไฟล์ด้วยฟังก์ชัน list_members() และตรวจสอบว่ามีข้อมูลหรือไม่ หากไม่มีสมาชิกอยู่ในระบบ โปรแกรมจะแจ้งว่า “No members in system” แล้วหยุดการทำงานทันที หากมีสมาชิกอยู่ ระบบจะแสดงรายชื่อทั้งหมดในรูปแบบตาราง โดยแสดงรายละเอียดของแต่ละคน เช่น ID, ชื่อ, เบอร์โทรศัพท์, ที่อยู่ และสถานะ (Active หรือ Deleted) เพื่อให้ผู้ใช้เลือกสมาชิกที่ต้องการแก้ไข จากนั้นผู้ใช้งานจะต้องกรอกหมายเลข ID ของสมาชิกที่ต้องการแก้ไข ระบบจะตรวจสอบว่า ID ที่ระบุมีอยู่จริงหรือไม่ หากไม่พบจะแสดงข้อความว่า “Member not found” และหยุดทำงาน เมื่อพบข้อมูลสมาชิกที่ต้องการแก้ไขแล้ว ระบบจะอนุญาตให้ผู้ใช้กรอกข้อมูลใหม่ในแต่ละช่อง ได้แก่ ชื่อ, เบอร์โทรศัพท์ และที่อยู่ โดยหากผู้ใช้ไม่กรอกค่าใหม่ ระบบจะเก็บค่าของเดิมไว้ (โดยใช้ข้อความว่า “Leave blank to keep current value.”) หลังจากกรอกข้อมูลเสร็จ โปรแกรมจะแสดงข้อมูลที่อัปเดตให้ผู้ใช้งานตรวจสอบอีกครั้ง และถามยืนยันก่อนบันทึก หากผู้ใช้ตอบ “y” หรือ “yes” ระบบจะทำการบันทึกข้อมูลใหม่ลงในไฟล์โดยใช้โครงสร้าง MEMBER_STRUCT.pack() พร้อมบันทึกเวลาแก้ไขล่าสุด (now_ts()) และแสดงข้อความ “Member updated successfully.” เพื่อยืนยันการแก้ไข หากไม่ยืนยัน ระบบจะยกเลิกการอัปเดตและแสดงข้อความ “Update canceled.”

```

def update_member():
    print("-- Update Member --")
    members = list_members()
    if not members:
        print("No members in system")
        return

    print_table(
        ["ID", "Name", "Phone", "Address", "Status"],
        [[m["id"], m["name"], m["phone"], m["addr"][:40], "Active" if m["active"] else "Deleted"] for m in members]
    )

    mid = safe_input("Enter Member ID to update: ", int)
    idx, mem = find_member_by_id(mid)
    if not mem:
        print("Member not found")
        return

    print("Leave blank to keep current value.")
    new_name = safe_input(f"Name [{mem['name']}]: ", allow_empty=True)
    new_phone = safe_input(f"Phone [{mem['phone']}]: ", allow_empty=True)
    new_addr = safe_input(f"Address [{mem['addr']}]: ", allow_empty=True)

    updated_name = new_name if new_name else mem["name"]
    updated_phone = new_phone if new_phone else mem["phone"]
    updated_addr = new_addr if new_addr else mem["addr"]

    print("\nPlease confirm the updated information:")
    print(f" ID : {mem['id']}")
    print(f" Name : {updated_name}")
    print(f" Phone : {updated_phone}")
    print(f" Addr : {updated_addr}")
    print(f" Status: {'Active' if mem['active'] else 'Deleted'}")

    confirm = input("Save changes? (y/n): ").strip().lower()
    if confirm not in ["y", "yes"]:
        print("Update canceled.")
        return

```

รูปภาพ 4-17 Update_member()

```

packed = MEMBER_STRUCT.pack(
    mem["id"],
    pack_fixed_str(updated_name, NAME_LEN),
    pack_fixed_str(updated_phone, PHONE_LEN),
    pack_fixed_str(updated_addr, ADDR_LEN),
    mem["active"],
    now_ts()
)
write_record_at(MEMBERS_FILE, MEMBER_STRUCT, idx, packed)
print("Member updated successfully.")

```

รูปภาพ 4-18 Update_member()

4.2.9 delete_member()

ฟังก์ชัน delete_member() ทำงานตามขั้นตอนดังนี้ ฟังก์ชันเริ่มต้นด้วยการแสดงข้อความ "-- Delete Member (soft delete) --" เพื่อแจ้งผู้ใช้งานที่กำลังลบสมาชิกแบบไม่ลบจริง จากนั้นให้ผู้ใช้งานกรอก Member ID ของสมาชิกที่ต้องการลบ ระบบจะตรวจสอบว่ามีสมาชิกที่ตรงกับ ID ที่กรอก

หรือไม่ หากไม่พบจะแจ้งว่า "Member not found" และหยุดทำงาน หากพบสมาชิกแล้ว ระบบจะตรวจสอบว่าสถานะสมาชิกยังเป็น Active อยู่หรือไม่ หากสมาชิกถูกลบไปแล้วจะแจ้งว่า "Member already deleted" และหยุดทำงาน ต่อมา ฟังก์ชันจะตรวจสอบว่ามีรายการยืมหนังสือที่ยังไม่คืนของสมาชิกนี้หรือไม่ หากพบว่ามีรายการยืมที่ยังไม่คืน ฟังก์ชันจะแจ้งว่าไม่สามารถลบสมาชิกได้ เพราะยังมีรายการยืมค้างอยู่ และหยุดทำงาน หากไม่มีรายการยืมค้าง ระบบจะทำการสร้างข้อมูลสมาชิกใหม่โดยเปลี่ยนสถานะเป็น inactive (0) และบันทึกข้อมูลนี้กลับไปยังไฟล์สมาชิกเดิม สุดท้ายระบบจะแจ้งผู้ใช้ว่าการลบสมาชิกสำเร็จด้วยความ "Member soft-deleted successfully."

```
def delete_member():
    print("-- Delete Member (soft delete) --")
    mid = safe_input("Member ID to delete: ", int)
    idx, mem = find_member_by_id(mid)
    if not mem:
        print("Member not found")
        return

    if mem["active"] == 0:
        print("Member already deleted")
        return

    # ตรวจสอบว่ามี loan ที่ยังไม่คืนหรือไม่
    active_loans = [l for l in list_loans() if l["member_id"] == mid and l["return_date"] == 0]
    if active_loans:
        print("Cannot delete member: still has active loans")
        return

    packed = MEMBER_STRUCT.pack(
        mem["id"],
        pack_fixed_str(mem["name"], NAME_LEN),
        pack_fixed_str(mem["phone"], PHONE_LEN),
        pack_fixed_str(mem["addr"], ADDR_LEN),
        0, # inactive
        now_ts()
    )
    write_record_at(MEMBERS_FILE, MEMBER_STRUCT, idx, packed)
    print("| Member soft-deleted successfully.")
```

รูปภาพ 4-19 delete_member()

4.2.10 view()

เมื่อเข้าสู่ฟังก์ชัน view() โปรแกรมจะเริ่มต้นด้วยการแสดงหัวข้อ “View Menu” เพื่อให้ผู้ใช้เลือกประเภทของข้อมูลที่ต้องการดู ซึ่งสามารถเลือกได้ระหว่าง Book, Member และ Loan โดยหากผู้ใช้พิมพ์เลข 0 โปรแกรมจะกลับไปยังเมนูหลักทันที แต่หากพิมพ์ประเภทที่ไม่ถูกต้อง โปรแกรมจะแจ้งเตือนว่าเป็นประเภทที่ไม่รู้จักและให้เลือกใหม่อีกครั้ง หลังจากเลือกประเภทข้อมูลแล้ว โปรแกรมจะแสดงโหมดการดูข้อมูลให้เลือก ซึ่งประกอบด้วย View All, View Filter และ View Single เพื่อให้ผู้ใช้เลือกรูปแบบการดูข้อมูลที่ต้องการ หากผู้ใช้เลือกโหมด View All โปรแกรมจะทำการแสดงข้อมูลทั้งหมดในหมวดที่เลือก เช่น หากเลือก Book จะเรียกฟังก์ชัน view_all_books()

เพื่อแสดงข้อมูลหนังสือทั้งหมดในระบบ หากเลือก Member จะเรียก `view_all_members()` เพื่อแสดงรายชื่อสมาชิกทั้งหมด และหากเลือก Loan จะเรียก `view_all_loans()` เพื่อแสดงประวัติการยืมทั้งหมด ข้อมูลจะถูกนำมาแสดงในรูปแบบตารางอย่างเป็นระเบียบ เพื่อให้ผู้ใช้สามารถดูภาพรวมได้อย่างชัดเจน แต่ถ้าผู้ใช้เลือกโหมด View Filter โปรแกรมจะให้ผู้ใช้กรอกเงื่อนไขเพื่อกรองข้อมูลเฉพาะส่วนที่ต้องการ เช่น หากเลือกดูข้อมูลหนังสือ (Book) จะสามารถกรอกคำค้นในชื่อเรื่อง ผู้แต่ง ปีพิมพ์ หรือสถานะของหนังสือได้ และหากไม่กรอกค่าหรือกด Enter โปรแกรมจะข้ามช่องนั้นไป จากนั้นจะแสดงเฉพาะหนังสือที่ตรงตามเงื่อนไขที่กำหนดในรูปแบบตารางพร้อมบอกจำนวนที่พบ ส่วนในกรณีของสมาชิก (Member) โปรแกรมจะให้กรอกคำค้นในชื่อ เบอร์โทรศัพท์ ที่อยู่ หรือสถานะของสมาชิก เพื่อกรองเฉพาะข้อมูลที่ต้องการดูเช่นกัน แต่สำหรับ Loan จะไม่มีการกรองข้อมูลและโปรแกรมจะแจ้งว่าการกรองไม่สามารถใช้ได้กับหมวดนี้ สุดท้าย หากเลือกโหมด View Single โปรแกรมจะให้ผู้ใช้กรอกหมายเลข ID ของรายการที่ต้องการดู เช่น Book ID, Member ID หรือ Loan ID จากนั้นโปรแกรมจะค้นหารายการนั้นและแสดงรายละเอียดทั้งหมดของข้อมูลรายการเดียวในรูปแบบตาราง โดยในกรณีของ Book จะแสดงข้อมูลชื่อเรื่อง ผู้แต่ง ปีพิมพ์ จำนวนเล่ม ทั้งหมด จำนวนที่ว่าง และสถานะ ในกรณีของ Member จะแสดงชื่อ เบอร์โทรศัพท์ ที่อยู่ และสถานะของสมาชิก ส่วน Loan จะแสดงข้อมูลรหัสการยืม รหัสหนังสือ รหัสสมาชิก วันที่ยืม วันที่คืน และสถานะการยืม

โดยสรุป ฟังก์ชัน `view()` เป็นฟังก์ชันที่ช่วยให้ผู้ใช้สามารถดูข้อมูลในระบบห้องสมุดได้อย่างละเอียดและยืดหยุ่น ไม่ว่าจะเป็นการดูข้อมูลทั้งหมด การกรองเฉพาะส่วนที่ต้องการ หรือการค้นหาข้อมูลเฉพาะรายการเดียว ซึ่งช่วยอำนวยความสะดวกในการจัดการและตรวจสอบข้อมูลต่าง ๆ ได้อย่างมีประสิทธิภาพ

```

def view():
    while True:
        print("\n-- View Menu --")
        print("\0 Back to Main Menu")
        t = safe_input("Type (book/member/loan): ", lambda s: s.lower())
        if t == "\0":
            break
        if t not in ["book", "member", "loan"]:
            print("Unknown type")
            continue

        print("Mode:")
        print("1) View All")
        print("2) View Filter")
        print("3) View Single")
        print("\0 Back")
        mode = safe_input("Select mode: ", lambda s: s.strip())
        if mode == "\0":
            continue # กลับไปเลือก Type ใหม่

        if mode == "1": # View All
            if t == "book":
                view_all_books()
            elif t == "member":
                view_all_members()
            elif t == "loan":
                view_all_loans()

        elif mode == "2": # Filter
            if t == "book":
                books = list_books(show_inactive=True)
                if not books:
                    print("No books found.")
                    continue
                kw_title = input("Keyword in Title (Enter to skip): ").strip().lower()
                kw_author = input("Keyword in Author (Enter to skip): ").strip().lower()
                kw_year = input("Year (Enter to skip): ").strip()
                kw_active = input("Status (active/deleted/Enter to skip): ").strip().lower()

                filtered_books = books

```

รูปภาพ 4-20 view()

```

        filtered_books = books
        if kw_title:
            filtered_books = [b for b in filtered_books if kw_title in b["title"].lower()]
        if kw_author:
            filtered_books = [b for b in filtered_books if kw_author in b["author"].lower()]
        if kw_year:
            filtered_books = [b for b in filtered_books if str(b["year"]) == kw_year]
        if kw_active == "active":
            filtered_books = [b for b in filtered_books if b["active"] == 1]
        elif kw_active == "deleted":
            filtered_books = [b for b in filtered_books if b["active"] == 0]

        headers = ["ID", "Title", "Author", "Year", "Avail", "Status"]
        rows = []
        for b in filtered_books:
            rows.append([b["id"], b["title"][:28], b["author"], b["year"], b["available"], "Active" if b["active"] else "Deleted"])
        print_table(headers, rows)
        print(f"Found {len(filtered_books)} book(s).")

    elif t == "member":
        members = list_members(show_inactive=True)
        if not members:
            print("No members found.")
            continue
        kw_name = input("Keyword in Name (Enter to skip): ").strip().lower()
        kw_phone = input("Keyword in Phone (Enter to skip): ").strip().lower()
        kw_addr = input("Keyword in Address (Enter to skip): ").strip().lower()
        kw_active = input("Status (active/deleted/Enter to skip): ").strip().lower()

```

รูปภาพ 4-21 view() ต่อ 1

```

        filtered_members = members
        if kw_name:
            filtered_members = [m for m in filtered_members if kw_name in m["name"].lower()]
        if kw_phone:
            filtered_members = [m for m in filtered_members if kw_phone in m["phone"].lower()]
        if kw_addr:
            filtered_members = [m for m in filtered_members if kw_addr in m["addr"].lower()]
        if kw_active == "active":
            filtered_members = [m for m in filtered_members if m["active"] == 1]
        elif kw_active == "deleted":
            filtered_members = [m for m in filtered_members if m["active"] == 0]

        headers = ["ID", "Name", "Phone", "Address", "Status"]
        rows = []
        for m in filtered_members:
            rows.append([m["id"], m["name"], m["phone"], m["addr"][:40], "Active" if m["active"] else "Deleted"])
        print_table(headers, rows)
        print(f"Found {len(filtered_members)} member(s).")

    else:
        print("Filter not available for loans")

elif mode == "3": # Single
    if t == "book":
        bid = safe_input("Book ID: ", int)
        _, b = find_book_by_id(bid)
        if not b:
            print("Book not found")
            continue
        headers = ["Field", "Value"]
        rows = [
            ["ID", b["id"]],
            ["Title", b["title"]],
            ["Author", b["author"]],
            ["Year", b["year"]],
            ["Total", b["total"]],
            ["Available", b["available"]],
            ["Active", "Yes" if b["active"] else "No"]
        ]
        print_table(headers, rows)

```

รูปภาพ 4-22 view() ต่อ 2

```

        elif t == "member":
            mid = safe_input("Member ID: ", int)
            _, m = find_member_by_id(mid)
            if not m:
                print("Member not found")
                continue
            headers = ["Field", "Value"]
            rows = [
                ["ID", m["id"]],
                ["Name", m["name"]],
                ["Phone", m["phone"]],
                ["Address", m["addr"]],
                ["Active", "Yes" if m["active"] else "No"]
            ]
            print_table(headers, rows)

        elif t == "loan":
            lid = safe_input("Loan ID: ", int)
            for idx, v in read_all_records(LOANS_FILE, LOAN_STRUCT):
                if v[0] == lid:
                    rid, book_id, member_id, borrow_ts, return_ts, active, last_mod = v
                    headers = ["Field", "Value"]
                    rows = [
                        ["Loan ID", rid],
                        ["Book ID", book_id],
                        ["Member ID", member_id],
                        ["Borrow Date", fmt_ts_full(borrow_ts)],
                        ["Return Date", fmt_ts_full(return_ts)],
                        ["Active", "Yes" if active else "No"]
                    ]
                    print_table(headers, rows)
                    break
            else:
                print("Loan not found")
        else:
            print("Invalid mode")

```

รูปภาพ 4-23 view() ต่อ 3

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงการระบบยืม-คืนหนังสือห้องสมุดที่พัฒนาขึ้น สามารถจัดการข้อมูลหนังสือ ข้อมูลสมาชิก และข้อมูลการยืม-คืนได้อย่างมีประสิทธิภาพ ระบบใช้การจัดเก็บข้อมูลแบบไฟล์ไบนารี พร้อมเมนูสำหรับเพิ่ม แก้ไข ลบ และแสดงผลข้อมูล นอกจากนี้ ระบบยังสามารถตรวจสอบสถานะของหนังสือได้ว่าอยู่ระหว่างการยืมหรือว่างอยู่ ควบคุมจำนวนเล่มที่ถูกยืม และสร้างรายงานสรุปผล เช่น หนังสือที่ถูกยืมมากที่สุด รายชื่อผู้ยืมปัจจุบัน และสถิติการใช้งานโดยรวม ทั้งนี้ช่วยให้การบริหารจัดการห้องสมุดสะดวก รวดเร็ว และลดความผิดพลาดจากการบันทึกแบบเอกสารกระดาษ

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

การพัฒนาโครงการพบปัญหาหลักคือความซับซ้อนในการจัดการไฟล์ไบนารี ซึ่งต้องใช้โครงสร้างข้อมูลคงที่ (struct) และอาจเกิดข้อผิดพลาดหากการเข้ารหัสหรือถอดรหัสไม่ถูกต้อง นอกจากนี้ การแสดงผลข้อมูลยังมีข้อจำกัด เช่น ความยาวของชื่อหนังสือหรือชื่อผู้ใช้ที่ต้องกำหนดตามขนาดที่กำหนด ระบบยังไม่มี การเชื่อมต่อฐานข้อมูลจริง ทำให้การจัดการข้อมูลจำนวนมากหรือการเข้าถึงพร้อมกันจากหลายผู้ใช้งานยังทำได้ไม่เต็มประสิทธิภาพ

5.3 ข้อเสนอแนะ

5.3.1 เพื่อให้ระบบสมบูรณ์และพร้อมใช้งานจริง ควรปรับปรุงดังนี้

5.3.2 พัฒนาระบบให้รองรับฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เช่น MySQL หรือ SQLite เพื่อรองรับข้อมูลจำนวนมากและการเข้าถึงหลายผู้ใช้งาน

5.3.3 เพิ่มฟังก์ชันค้นหาและกรองข้อมูล เช่น ค้นหาหนังสือตามชื่อ ผู้แต่ง หรือปีที่พิมพ์

5.3.4 พัฒนาเป็นโปรแกรมผู้ใช้แบบกราฟิก (GUI) เพื่อความสะดวกในการใช้งานจริง

5.4 ประโยชน์ที่ผู้จัดทำได้รับ

ผู้จัดทำโครงการได้รับความรู้และประสบการณ์ด้านการออกแบบระบบ การเขียนโปรแกรมด้วยภาษา Python การใช้โครงสร้างข้อมูลแบบไบนารี รวมถึงการคิดวิเคราะห์และแก้ไขปัญหาเชิง

ตรรกะ นอกจากนี้ ยังได้ฝึกทักษะการทำงานเป็นทีม การแบ่งหน้าที่รับผิดชอบ และการจัดการเวลาให้สอดคล้องกับแผนงาน ทำให้ผู้จัดทำมีความเข้าใจในกระบวนการพัฒนาระบบซอฟต์แวร์มากยิ่งขึ้น