ระบบยืม – คืนหนังสือห้องสมุด

Borrow and return library books

นางสาวกนกวรรณ ศรีประไชย	**รหัส 6806022510041**	**SEC 2**
นายนันทวัฒน์ นันทวิสาร	**รหัส 6806022510050**	**SEC 2**
นายบุญญฤทธิ์ วุฒิ	**รหัส 6806022510173**	**SEC 2**
นายกฤษนัย โถมสันเทียะ	**รหัส 6806022510203**	**SEC 2**

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2568

คำนำ

การจัดทำโครงงาน “ระบบยืม - คืนหนังสือห้องสมุด” นี้เป็นส่วนหนึ่งของวิชา Computer Programming ของหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ เพื่อให้นักศึกษาได้นำความรู้ที่เรียนมาทั้งหมดมาประยุกต์ใช้ในการพัฒนาโปรแกรมที่สามารถทำงานได้จริง โดยเน้นการออกแบบและเขียนโปรแกรมด้วยภาษา Python ซึ่งเป็นภาษาที่เรียนมาในวิชา Computer Programing โดยโครงงานนี้จะช่วยการคิดวิเคราะห์และการแก้ปัญหาทางเทคนิค เพื่อเตรียมความพร้อมในการประกอบอาชีพด้านวิศวกรรมสารสนเทศและเครือข่ายในอนาคต

คณะผู้จัดทำหวังว่า รายงานฉบับนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักเรียน นักศึกษา ที่กำลังหาข้อมูลเรื่องนี้อยู่ หากมีข้อแนะนำหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้และขอ อภัยมา ณ ที่นี้ด้วย

สารบัญ

หน้า

สารบัญ(ต่อ)

สารบัญรูปภาพ

สารบัญรูปภาพ(ต่อ)

สารบัญตาราง

บทที่ 1
บทนำ

1.1 วัตถุประสงค์ของโครงงาน

1.1.1 เพื่อพัฒนาระบบยืม – คืนหนังสือห้องสมุดได้อย่างมีประสิทธิภาพ

1.1.2 เพื่อฝึกฝนทักษะการเขียนโปรแกรมด้วย Python

1.1.3 เพื่อเรียนรู้วิธีการจัดการข้อมูลและไฟล์

1.1.4 เพื่อเรียนรู้การทำางานร่วมกันเป็นทีม

1.2 ขอบเขตของโครงงาน

1.2.1 ระบบยืม – คืนหนังสือห้องสมุดมีฟังก์ชั่นพื้นฐานทั้งหมด 15 ฟังก์ชั่น เช่น

1. เพิ่มหนังสือ
2. แก้ไขหนังสือ
3. ดูข้อมูลหนังสือ
4. ลบหนังสือ
5. กลับไปที่เมนู
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 4 ไฟล์ ได้แก่

1. แฟ้มข้อมูลหนังสือ books.dat
2. แฟ้มข้อมูลสมาชิก members.dat
3. แฟ้มข้อมูลการยืม-คืน loans.dat
4. ไฟล์ report.txt

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

1.2.4 ระบบยืม - คืนหนังสือห้องสมุดจะมีเมนูเพื่อให้ผู้ใช้สามารถเลือก ดำเนินการได้

1.3 ประโยชน์ที่ได้รับ

1.3.1 พัฒนาระบบที่สามารถทำการยืม - คืนหนังสือได้อย่างมีประสิทธิภาพ

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
ระบบยืม - คืนหนังสือห้องสมุด

2.1 แฟ้มข้อมูลหนังสือ books.dat

แฟ้มข้อมูลหนังสือประกอบด้วย 8 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด(bytes)	ตัวอย่าง
book_id	I	5	รหัสหนังสือ
title	50s	50	ชื่อหนังสือ
status	I	4	สถานะ Active=1, Deleted=0
author	30s	50	ชื่อผู้แต่ง
year	I	4	ปีที่พิมพ์
copies	I	3	จำนวนเล่ม
created_at	I	4	timestamp สร้าง record
updated_at	I	4	timestamp สร้าง record

ตารางที่ 2.1 แฟ้มข้อมูลหนังสือ

2.1.1 book_id รหัสหนังสือ

Book_id เป็นรหัสหนังสือที่ใช้ในการระบุหนังสือแต่ละเล่มอย่างชัดเจนและไม่ซ้ำกัน ฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของตัวเลข (integer) เช่น 1001, 1002, 1003 เป็นต้น การมีรหัสหนังสือที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อหลีกเลี่ยงความสับสนระหว่างหนังสือหลายเล่ม และช่วยให้สามารถค้นหาและเรียกดูข้อมูลของหนังสือได้อย่างแม่นยำและรวดเร็ว โดยเฉพาะในกรณีที่มีหนังสือจำนวนมาก

2.1.2 title ชื่อหนังสือ

title คือ ชื่อเต็มของหนังสือแต่ละเล่ม ซึ่งฟิลด์นี้จะแสดงข้อมูลชื่อหนังสือแต่ละเล่มของห้องสมุด ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (string) ตัวอย่างเช่น " PATRIOT " หรือ " COMPUTER SYSTEMS " การมีชื่อหนังสือในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการเรียกดูข้อมูล ตรวจสอบการยืม – คืนและทำการแก้ไขข้อมูลต่างๆ หนังสือแต่ละเล่มจะมีชื่อตามที่ระบุในการลงทะเบียน และระบบจะใช้ชื่อดังกล่าวในการค้นหาและแสดงผลข้อมูลที่เกี่ยวข้องกับหนังสือเล่มนั้น

2.1.3 status สถานะ

status คือ สถานะหนังสือเล่มนั้นๆ ซึงฟิลด์นี้จะแสดงข้อมูลสถานะของหนังสือแต่ละเล่ม ฟิลด์นี้เป็นประเภทข้อมูลตัวเลขจำนวนเต็ม (integer) เช่น 1 = Active (ใช้งานอยู่/ยังมีในระบบ) หรือ 0 = Deleted (ถูกลบ/ไม่ใช้งานแล้ว) การมีสถานะของหนังสือในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการดูสถานะของหนังสือแต่ละเล่ม และทำการแก้ไขข้อมูลต่างๆ หนังสือแต่ละเล่มจะมีสถานะตามที่ระบุในการใช้งานอยู่ หรือการถูกลบ และ ระบบจะใช้สถานะดังกล่าวในการแสดงผลข้อมูลที่เกี่ยวข้องกับหนังสือเล่มนั้น

2.1.4 author ชื่อผู้แต่ง

author คือ ชื่อของผู้เขียนหนังสือเล่มนั้นๆ ซึงฟิลด์นี้จะแสดงข้อมูลชื่อผู้เขียนหนังสือแต่ละเล่ม ฟิลด์นี้เป็นประเภทข้อมูลข้อความ (string) ตัวอย่างเช่น " ALEXEI NAVALNY " หรือ " RANDAL E. BRYANT " การมีชื่อผู้เขียนหนังสือในระบบมีความสำคัญอย่างยิ่ง เพราะใช้ในการเรียกดูข้อมูล ตรวจสอบหนังสือ การยืม - คืน และทำการแก้ไขข้อมูลต่างๆ หนังสือแต่ละเล่มจะมีชื่อตามที่ระบุในการลงทะเบียน และ ระบบจะใช้ชื่อดังกล่าวในการค้นหาและแสดงผลข้อมูลที่เกี่ยวข้องกับหนังสือเล่มนั้น

2.1.5 year ปีที่พิมพ์

year คือ ปีที่เขียนหนังสือที่ใช้ในการระบุหนังสือแต่ละเล่มอย่างชัดเจน ฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของตัวเลข (integer) เช่น 2025, 2019, 2016 เป็นต้น การมีปีที่เขียนหนังสือที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อแสดงปีที่เขียนหนังสือเล่มนั้นๆ และช่วยให้สามารถทราบปีที่เขียนได้และทราบระยะเวลาของหนังสือเพื่อเพิ่มความน่าเชื่อถือได้มากยิ่งขึ้น

2.1.6 copies จำนวนเล่ม

copies ใช้เก็บจำนวนเล่มของหนังสือที่มีอยู่ในห้องสมุดหรือคลังหนังสือ ฟิลด์นี้เก็บในรูปแบบตัวเลขจำนวนเต็ม (integer) เช่น 3, 10, 25 การเก็บจำนวนเล่มจะช่วยในการบริหารจัดการทรัพยากร เช่น การยืม–คืน และการตรวจสอบว่าหนังสือเล่มนั้นมีเพียงพอต่อการใช้งานหรือไม่

2.1.7 created_at timestamp สร้าง record

created_at ใช้เก็บเวลาในรูปแบบตัวเลข (timestamp) ที่บันทึกว่ารายการหนังสือนี้ถูกสร้างขึ้นเมื่อใด เช่น 1727767200 ซึ่งสอดคล้องกับวันที่และเวลาจริง การมีฟิลด์นี้ช่วยให้สามารถตรวจสอบและติดตามการบันทึกข้อมูลได้ว่าหนังสือถูกเพิ่มเข้าสู่ระบบตั้งแต่เมื่อไร

2.1.8 updated_at timestamp สร้าง record

updated_at ใช้เก็บเวลาในรูปแบบตัวเลข (timestamp) ที่บันทึกว่ามีการแก้ไขข้อมูลหนังสือครั้งล่าสุดเมื่อใด เช่น 1727853600 การเก็บข้อมูลนี้ช่วยให้ผู้ดูแลระบบสามารถตรวจสอบการอัพเดตข้อมูลหนังสือ และติดตามการเปลี่ยนแปลงที่เกิดขึ้นในระบบได้อย่างถูกต้อง

2.2 แฟ้มข้อมูลสมาชิก members.dat

แฟ้มข้อมูลสมาชิกประกอบด้วย 7 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด(bytes)	คำอธิบาย
member_id	I	5	รหัสสมาชิก
status	I	4	Active=1, Deleted=0
name	50s	50	ชื่อ – นามสกุล
birth_year	I	4	ปีเกิด
max_loan	I	4	จำนวนหนังสือสูงสุดที่ยืมได้
created_at	I	4	timestamp
updated_at	I	4	timestamp

ตารางที่ 2.1 แฟ้มข้อมูลสมาชิก

2.2.1 member_id รหัสสมาชิก

member_id เป็นรหัสที่ใช้ระบุสมาชิกแต่ละคนในระบบ ฟิลด์นี้เป็นตัวเลขจำนวนเต็ม (integer) มีความเป็นเอกลักษณ์ไม่ซ้ำกัน เช่น 10001, 20015 การมีรหัสสมาชิกช่วยให้ระบบสามารถจัดการข้อมูลสมาชิกจำนวนมากได้อย่างถูกต้องและรวดเร็ว

2.2.2 status สถานะ

status ใช้เก็บสถานการณ์ใช้งานของสมาชิก ในรูปแบบข้อมูลตัวเลขจำนวนเต็ม (integer) เช่น 1 = Active (ใช้งานอยู่/ยังมีในระบบ) หรือ 0 = Deleted (ถูกลบ/ไม่ใช้งานแล้ว) ฟิลด์นี้ช่วยให้ระบบสามารถจัดการแยกแยะสมาชิกที่ยังอยู่กับสมาชิกที่ไม่ใช้งานแล้วได้อย่างมีประสิทธิภาพ

2.2.3 name ชื่อสมาชิก

name เป็นฟิลด์ข้อความ (string) ความยาวสูงสุด 50 ตัวอักษร ใช้เก็บชื่อ–นามสกุลของสมาชิก เช่น "สมชาย ใจดี", "John Smith" ฟิลด์นี้เป็นข้อมูลสำคัญเพื่อแสดงผลและยืนยันตัวตนของสมาชิก

2.2.4 birth_year ปีเกิด

birth_year เป็นตัวเลขจำนวนเต็ม (integer) ใช้เก็บปีเกิดของสมาชิก เช่น 1999, 2005 ข้อมูลนี้อาจถูกนำไปใช้ในการตรวจสอบอายุ การจัดกลุ่มสมาชิกตามช่วงวัย หรือใช้กำหนดสิทธิพิเศษบางอย่างสำหรับสมาชิกแต่ละช่วงอายุ

2.2.5 max_loan จำนวนหนังสือสูงสุดที่ยืมได้

max_loan เป็นฟิลด์จำนวนเต็ม (integer) ที่กำหนดจำนวนสูงสุดของหนังสือที่สมาชิกแต่ละคนสามารถยืมได้ เช่น 3, 5, 10 ช่วยควบคุมการยืมหนังสือเพื่อป้องกันไม่ให้มีการยืมเกินขีดจำกัดที่ระบบกำหนด

2.2.6 created_at timestamp สร้าง record

created_at เป็นฟิลด์แบบ timestamp (เก็บในรูปตัวเลข integer 4 bytes) ใช้ระบุวันที่และเวลาที่ข้อมูลสมาชิกถูกสร้างขึ้นในระบบ เช่น 1727767200 (2024-10-01 10:00:00) ข้อมูลนี้ช่วยให้สามารถตรวจสอบประวัติการเพิ่มสมาชิกได้

2.2.7 updated_at timestamp สร้าง record

updated_at เป็นฟิลด์แบบ timestamp (integer 4 bytes) ใช้เก็บวันที่และเวลาที่มีการแก้ไขข้อมูลสมาชิกครั้งล่าสุด เช่น 1727853600 (2024-10-02 10:00:00) ฟิลด์นี้มีประโยชน์ในการติดตามการเปลี่ยนแปลงข้อมูลและตรวจสอบความถูกต้องของข้อมูลล่าสุดในระบบ

2.3 แฟ้มข้อมูลการยืม-คืน loans.dat

แฟ้มข้อมูลสมาชิกประกอบด้วย 8 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด(bytes)	คำอธิบาย
ts	I	4	timestamp เหตุการณ์ (เวลาที่เกิดการยืมหรือคืน)
op_code	I	4	1=ADD,2=UPDATE,3=DELETE,4=VIEW
book_id	I	4	รหัสหนังสือ (อ้างอิงไปยัง books.dat)
member_id	I	4	รหัสสมาชิก (อ้างอิงไปยัง members.dat)
loan_date	10s	10	วันที่ยืม (YYYY-MM-DD)
return_date	10s	10	วันที่คืน
status_after	I	4	สถานะของ record หลังเหตุการณ์ เช่น 1=ยืมอยู่, 0=คืนแล้ว
is_rented_after	I	4	แสดงสถานะว่าหนังสือเล่มนี้ถูกยืมอยู่หรือไม่ (1=กำลังถูกยืม, 0=ว่าง)

ตารางที่ 2.3 แฟ้มข้อมูลการยืม-คืน

2.3.1 ts timestamp เหตุการณ์ (เวลาที่เกิดการยืมหรือคืน)

เป็นฟิลด์แบบ timestamp (integer 4 bytes) ใช้เก็บวันที่และเวลาที่เกิดเหตุการณ์ เช่น การยืมหรือคืนหนังสือ ข้อมูลนี้ช่วยให้สามารถบันทึกลำดับเวลาของเหตุการณ์ได้อย่างถูกต้อง และใช้ตรวจสอบย้อนหลังได้

2.3.2 op_code รหัสดำเนินการ

เป็นฟิลด์จำนวนเต็ม (integer 4 bytes) ใช้ระบุประเภทของการทำงาน เช่น 1 = ADD (เพิ่มข้อมูล) 2 = UPDATE (แก้ไขข้อมูล) 3 = DELETE (ลบข้อมูล) 4 = VIEW (เรียกดูข้อมูลฟิลด์นี้ช่วยบอกว่าการเปลี่ยนแปลงในระบบเกิดจากการกระทำแบบใด

2.3.3 book_id รหัสหนังสือ (อ้างอิงไปยัง books.dat)

book id เป็นรหัสหนังสือที่ใช้ในการระบุหนังสือแต่ละเล่มอย่างชัดเจนและไม่ซ้ำกัน ฟิลด์นี้ถูกสร้างขึ้นโดยระบบในรูปแบบของตัวเลข (integer) เช่น 1001, 1002, 1003 เป็นต้น การมีรหัสหนังสือที่เป็นเอกลักษณ์นี้เป็นสิ่งจำเป็นเพื่อหลีกเลี่ยงความสับสนระหว่างหนังสือหลายเล่ม และช่วยให้สามารถค้นหาและเรียกดูข้อมูลของหนังสือได้อย่างแม่นยำและรวดเร็ว โดยเฉพาะในกรณีที่มีหนังสือจำนวนมาก

2.3.4 member_id รหัสสมาชิก (อ้างอิงไปยัง members.dat)

เป็นฟิลด์จำนวนเต็ม (integer 4 bytes) ใช้เก็บรหัสสมาชิกที่เกี่ยวข้องกับเหตุการณ์ โดยอ้างอิงไปยังข้อมูลในไฟล์ members.dat ช่วยให้ทราบว่าใครเป็นผู้ทำรายการยืม–คืน

2.3.5 loan_date วันที่ยืม

เป็นฟิลด์ข้อความ (string ขนาด 10 ตัวอักษร) ใช้เก็บวันที่ยืมหนังสือในรูปแบบ YYYY-MM-DD เช่น 2025-09-30 ข้อมูลนี้ใช้ตรวจสอบว่าหนังสือถูกยืมไปตั้งแต่เมื่อใด

2.3.6 return_date วันที่คืน

เป็นฟิลด์ข้อความ (string ขนาด 10 ตัวอักษร) ใช้เก็บวันที่คืนหนังสือในรูปแบบ YYYY-MM-DD เช่น 2025-10-05 หากยังไม่ได้คืนค่าของฟิลด์นี้อาจเป็นค่าว่าง

2.3.6 status_after สถานะของ record หลังเหตุการณ์

เป็นฟิลด์จำนวนเต็ม (integer 4 bytes) ใช้เก็บสถานะของรายการหลังจากเหตุการณ์เกิดขึ้น เช่น 1 = ยืมอยู่ 0 = คืนแล้ว ฟิลด์นี้ช่วยบันทึกผลลัพธ์ของเหตุการณ์ยืมหรือคืนอย่างชัดเจน

2.3.8 is_rented_after แสดงสถานะว่าหนังสือเล่มนี้ถูกยืมอยู่หรือไม่

เป็นฟิลด์จำนวนเต็ม (integer 4 bytes) ใช้บอกสถานะว่าหนังสือเล่มนั้นหลังเหตุการณ์ยังคงถูกยืมอยู่หรือไม่ เช่น 1 = กำลังถูกยืม 0 = ว่าง/พร้อมให้ยืม ฟิลด์นี้มีประโยชน์ในการตรวจสอบสภาพล่าสุดของหนังสือโดยตรง

2.4 ไฟล์ report.txt

ไฟล์ report.txt ในระบบยืม - คืนหนังสือห้องสมุดของคุณประกอบด้วย 7 ฟิลด์หลัก ซึ่ง แต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

```
Library Borrow System - Summary Report
Generated At : 2025-09-24 19:06 (+0700)
App Version : 2.0
Encoding : UTF-8 (fixed-length)

-----------------------------------------------------------------------------------------------------------------------
BookID | Title                                    | Author                  | Year | Copies | Borrowed By         | Status
-----------------------------------------------------------------------------------------------------------------------
1001   | COMPUTER NETWORKS AND INTERNETS           | DOUGLAS E. COMER        | 2015 | 10     | 1.Lucas             | Active
       |                                          |                         |      |        | 2.Matthew           |
1002   | IOS PROGRAMMING                          | CHRISTIAN KEUR          | 2016 | 10     | 0                   | Active
1003   | AI-ASSISTED PROGRAMMING                  | TOM TAULLI              | 2024 | 12     | 1.Steven            | Active
1004   | INTRODUCTION TO PROGRAMMING WITH JAVA    | JOHN DEAN               | 2013 | 20     | 0                   | Active
1005   | MUST KNOW HIGH SCHOOL COMPUTER PROGRAMMING | JULIE SWAY              | 2020 | 15     | 1.Daniel            | Active
1006   | DIGITAL MAVERICKS                        | DEBBIE SOON             | 2025 | 10     | 1.Sofia             | Active
1007   | CONCEPTS OF PROGRAMMING LANGUAGES        | SEBESTA                 | 2015 | 12     | 1.Sofia             | Active
1008   | BAYESIAN METHODS FOR HACKERS             | CAMERON DAVIDSON-PILON  | 2015 | 20     | 0                   | Active
1009   | C PROGRAMMING ABSOLUTE BEGINNER'S GUIDE  | PERRY                   | 2013 | 10     | 1.Thomas            | Active
1010   | R PROGRAMMING FOR MASS SPECTROMETRY      | RANDALL K. JULIAN       | 2025 | 18     | 0                   | Active
1011   | DATABASE SYSTEMS                         | THOMAS CONNOLLY         | 2015 | 25     | 1.Lucas             | Active
       |                                          |                         |      |        | 2.Olivia            |
       |                                          |                         |      |        | 3.Michael           |
1012   | MODERN DATABASE MANAGEMENT               | HOFFER                  | 2016 | 15     | 0                   | Active
1013   | PATRIOT                                  | ALEXEI NAVALNY          | 2025 | 20     | 1.Matthew           | Active
1014   | PLUCKED                                  | MARYN MCKENNA           | 2019 | 10     | 1.Henry             | Active
       |                                          |                         |      |        | 2.Olivia            |
1015   | COMPUTER SYSTEMS                         | RANDAL E. BRYANT        | 2016 | 15     | 0                   | Active
1016   | ENGLISH LANGUAGE SYSTEMS                 | NARUTHAI                | 2025 | 20     | 1.Daniel            | Active
1017   | DIGITAL SAT STUDY GUIDE                  | BRIAN W. STEWART        | 2025 | 16     | 0                   | Active
1018   | BUSINESS PARTNER A1                      | MARGARET O'KEEFFE       | 2021 | 20     | 1.Daniel            | Active
1019   | COMPUTER APPLICATION                     | JONATHAN CHAN           | 2015 | 15     | 0                   | Active
1020   | SWIFT FOR PROGRAMMERS                    | PAUL DEITEL             | 2015 | 20     | 1.Michael           | Active

Summary (Active Books Only)
- Total Books   : 20
- Active Books  : 20
- Deleted Books : 0
- Borrowed Now  : 16
- Available Now : 297

Borrow Statistics (Active only)
- Most Borrowed Book : DATABASE SYSTEMS (3 times)
- Currently Borrowed : 16
- Active Members     : 10
```

รูปภาพที่ 2-1 ไฟล์ report

2.4.1 header_text ส่วนหัวรายงาน

เป็นฟิลด์ข้อความ (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" ฟิลด์นี้มีประโยชน์ในการตรวจสอบว่าไฟล์รายงานถูกสร้างด้วยเวอร์ชันของระบบใด

2.4.4 encoding การเข้ารหัสไฟล์

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

2.4.5 book_table_header หัวตาราง

book_table_header เป็นฟิลด์ข้อความ (string 80 bytes) ใช้เก็บหัวตารางสำหรับแสดงข้อมูลหนังสือ เช่น "BookID | Title | Author | Year | Copies | Borrowed By | Status" เพื่อกำหนดโครงสร้างของตารางในส่วนรายงาน

2.4.6 book_records ข้อมูลตาราง

เป็นฟิลด์ข้อความ (string 120 * N bytes, โดย N = จำนวนหนังสือ) ใช้เก็บข้อมูลหนังสือแต่ละเล่มในรูปแบบความยาวคงที่ (fixed-length record) เช่น รายการรหัสหนังสือ, ชื่อเรื่อง, ผู้แต่ง, ปีพิมพ์, จำนวนเล่ม สมาชิกที่ยืมหนังสือ และสถานะหนังสือ

2.4.6 summary_section สรุปข้อมูล

เป็นฟิลด์ข้อความ (string 150 bytes) ใช้เก็บข้อมูลสรุปของระบบ เช่น Total = จำนวนหนังสือทั้งหมด Active = หนังสือที่ใช้งานอยู่ Deleted = หนังสือที่ถูกลบ Borrowed = หนังสือที่ถูกยืม Available = หนังสือที่ว่าง ช่วยให้ผู้ใช้เห็นภาพรวมของฐานข้อมูลหนังสือได้อย่างรวดเร็ว

2.4.7 statistics_section สถิติการยืม

เป็นฟิลด์ข้อความ (string 150 bytes) ใช้เก็บข้อมูลสถิติการยืม เช่น Most Borrowed = หนังสือที่ถูกยืมมากที่สุด Currently Borrowed = จำนวนหนังสือที่กำลังถูกยืม Active Members = จำนวนสมาชิกที่มีการยืมอยู่ในปัจจุบัน ฟิลด์นี้มีประโยชน์ต่อการวิเคราะห์พฤติกรรมการใช้งานของสมาชิกและการวางแผนจัดการหนังสือ

บทที่ 3
การใช้งานระบบยืม – คืนหนังสือห้องสมุด

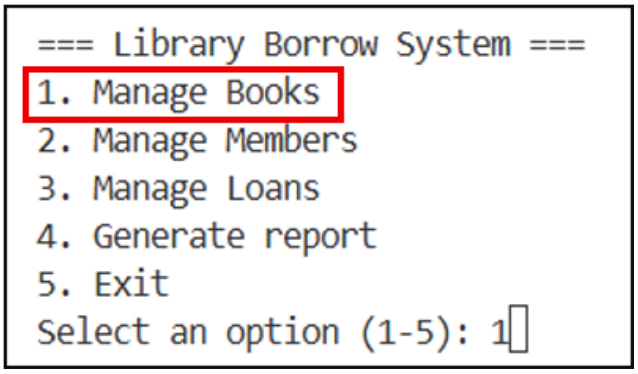
โปรแกรมการยืม – คืนหนังสือห้องสมุดคือการช่วยการยืม - คืนหนังสือให้สะดวกและง่ายขึ้น และยังช่วยจัดประเภทของหนังสือให้ดูง่ายขึ้นโดยการช่วยทำรายงานไปเก็บไว้ยังไฟล์Text

โปรแกรมการยืม - คือหนังสือห้องสมุดประกอบไปด้วย การเพิ่มข้อมูลที่ จะเก็บข้อมูล Book ID, Title, Author, Year , copies, borrowed by, Status แสดงข้อมูลหนังสือทั้งหมด ในโปรแกรมค้นหาข้อมูลโดยใช้ Book ID และ Title ในการค้นหาอัพเดทข้อมูลสามารถเปลี่ยนแปลงข้อมูลได้ แต่ถ้าไม่ต้องการแก้ไขข้อมูลบางส่วนสามารถกด Enter เพื่อข้ามการเปลี่ยนแปลงในข้อมูลนั้นๆ ได้เลยลบข้อมูลโดยการใช้เลข Book ID เพื่อลบข้อมูลทั้งหมดของเลข Book ID นั้น สร้างรายงานเพื่อทำสรุปการยืม – คืนหนังสือที่มีข้อมูลทั้งหมดในโปรแกรมจบการทำงานของโปรแกรม

สำหรับผู้ใช้งานโปรแกรม

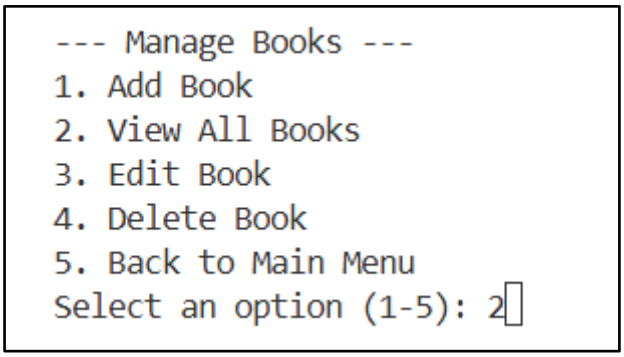
3.1 การใช้งานโปรแกรมระบบยืม - คืนห้องสมุด

3.1.1 กรอกหมายเลข 1 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Manage Books เพิ่มข้อมูลที่ประกอบไปด้วย Add Book, View All Books, Edit Book, Delete Book, Back to Main Menu



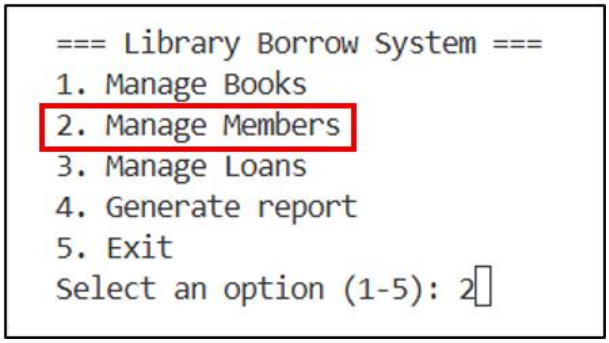
รูปภาพที่ 3- 1 การเลือกใช้งานฟังก์ชั่น Book

3.1.2 เมื่อเมนูฟังก์ชั่น Manage Book ขึ้นมาแล้วจากนั้นก็สามารถระบุเมนูที่ต้องการเลือกได้



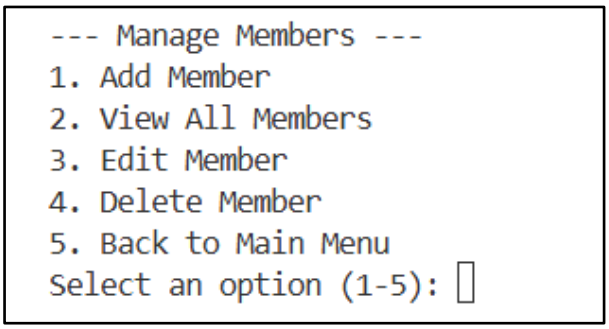
รูปภาพที่ 3- 2 เมนูของ Book

3.1.3 กรอกหมายเลข 2 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Manage Members เพิ่มข้อมูลที่ประกอบไปด้วย Add Member, View All Members, Edit Member, Delete Member, Back to Main Menu



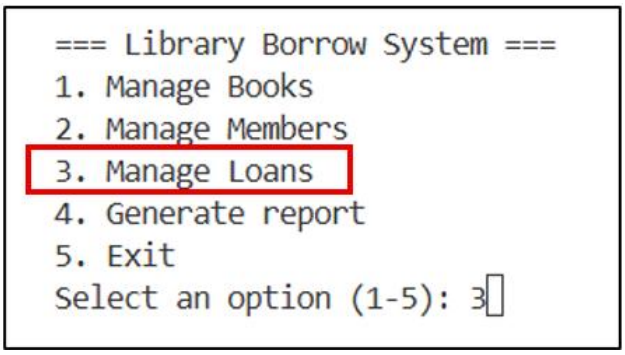
รูปภาพที่ 3- 3 การเลือกใช้งานฟังก์ชั่นของ Members

3.1.4 เมื่อเมนูฟังก์ชั่น Manage Members ขึ้นมาแล้วจากนั้นก็สามารถระบุเมนูที่ต้องการเลือกได้



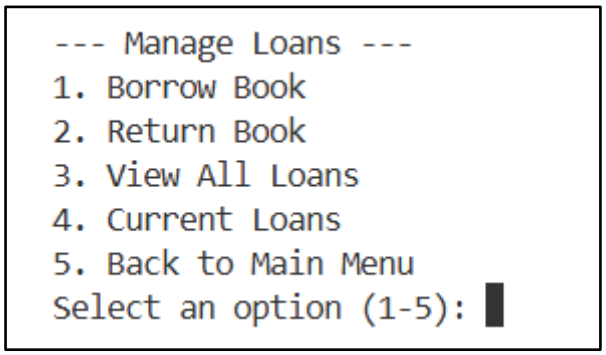
รูปภาพที่ 3- 4 เมนูของ Members

3.1.5 กรอกหมายเลข 3 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Manage Loans เพิ่มข้อมูลที่ประกอบไปด้วย Borrow Book, Return Book, View All Lons, Current Loans, Back to Main Menu



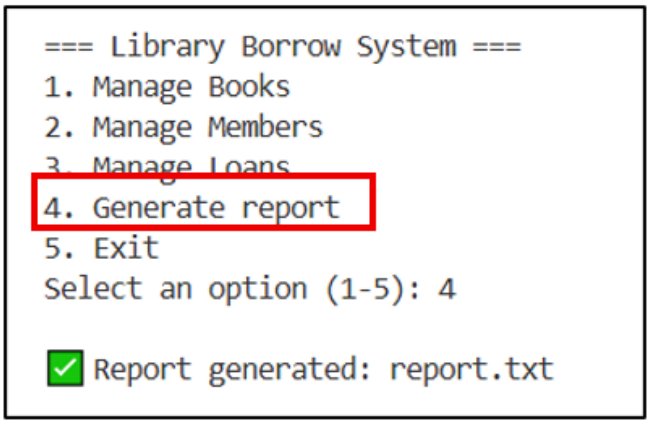
รูปภาพที่ 3- 5 การเลือกใช้งานฟังก์ชั่นของ Loans

3.1.6 เมื่อเมนูฟังก์ชั่น Manage Loans ขึ้นมาแล้วจากนั้นก็สามารถระบุเมนูที่ต้องการเลือกได้



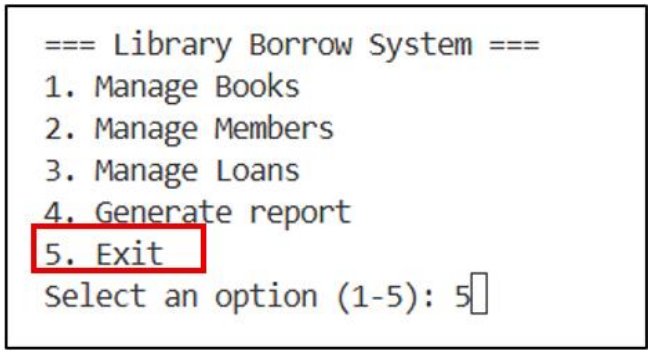
รูปภาพที่ 3- 6 เมนูของ Loans

3.1.7 กรอกหมายเลข 4 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Generate report เพื่อเพิ่มไฟล์ report.txt ที่สามารถเขียน report ถึงห้องสมุดได้



รูปภาพที่ 3- 7 การเลือกใช้งานฟังก์ชั่นของ Loans

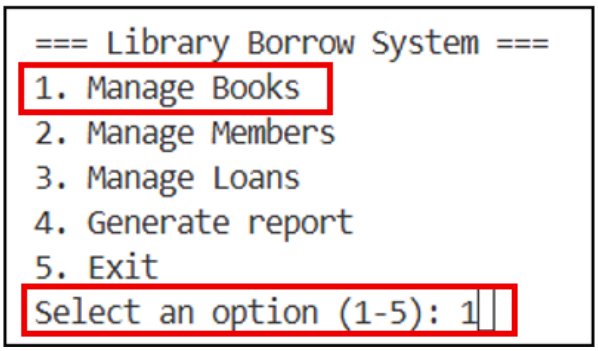
3.1.8 กรอกหมายเลข 5 ภายในกรอบสีแดงเพื่อเรียกฟังก์ชัน Exit เพื่อออกจากโปรแกรม



รูปภาพที่ 3- 8 การเลือกใช้งานฟังก์ชั่นของ Exit

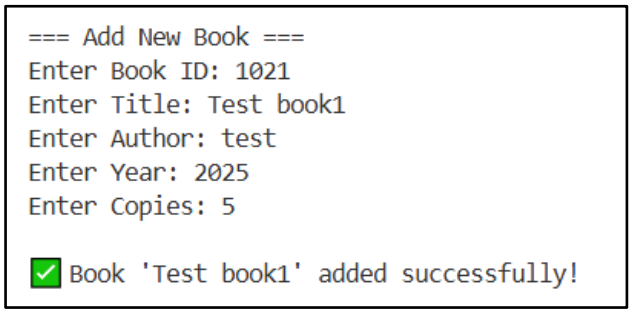
3.2 การใช้งานโปรแกรมเพิ่มข้อมูล

3.2.1 กรอกหมายเลข 1 เพื่อเพิ่มข้อมูลทั้งหมดของหนังสือที่มีในโปรแกรม



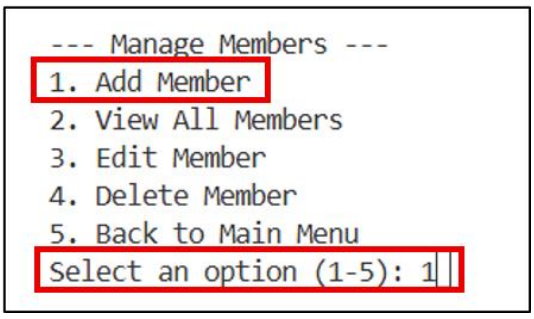
รูปภาพที่ 3- 9 การเลือกใช้งานฟังก์ชั่น Add Book

3.2.2 เมื่อกดเลือกหมายเลข 1 จะปรากฏหัวข้อการใส่รหัสหนังสือและจากนั้นใส่ข้อมูลในหัวข้อทั้งหมดดังภาพที่ 3-10



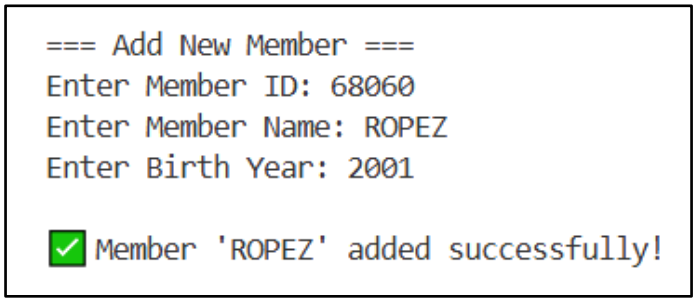
รูปภาพที่ 3- 10 การเพิ่มหนังสือ

3.2.3 กรอกหมายเลข 1 เพื่อเพิ่มข้อมูลทั้งหมดของผู้ใช้งานที่มีในโปรแกรม



รูปภาพที่ 3- 11 การเลือกใช้งานฟังก์ชั่น Add Member

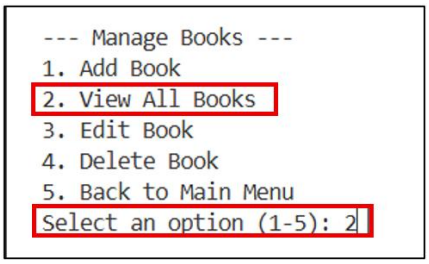
3.2.4 เมื่อกดเลือกหมายเลข 1 จะปรากฏหัวข้อการใส่รหัสผู้ใช้งานและจากนั้นใส่ข้อมูลในหัวข้อทั้งหมดดังภาพที่ 3-12



รูปภาพที่ 3- 12 การเพิ่มผู้ใช้งาน

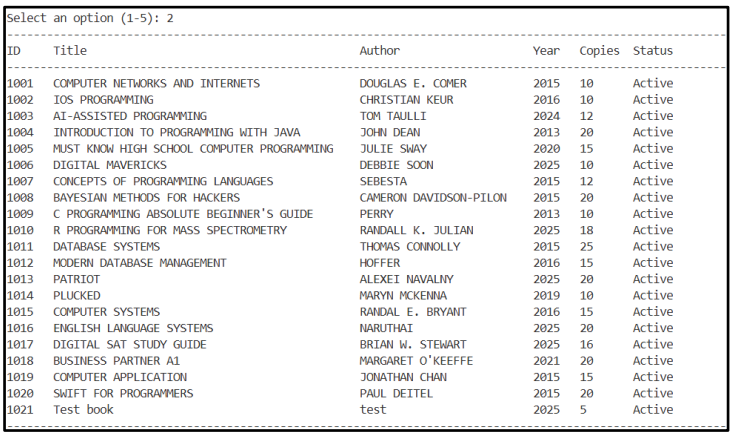
3.3 การใช้งานโปรแกรมแสดงข้อมูล

3.3.1 กรอกหมายเลข 2 เพื่อแสดงข้อมูลทั้งหมดของหนังสือที่มีในโปรแกรม



รูปภาพที่ 3- 13 การเลือกใช้งานฟังก์ชั่น View All Book

3.3.2 เมื่อกดเลือกหมายเลข 2 จะปรากฏข้อมูลหนังสือทั้งหมดดังภาพที่ 3-14

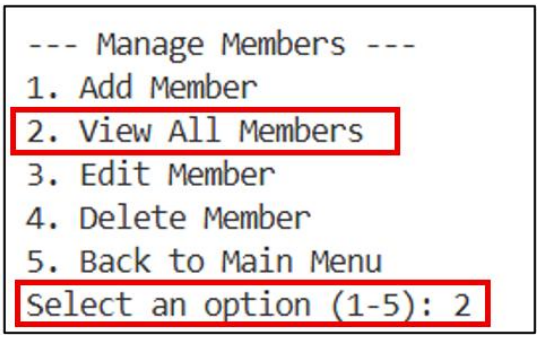


Select an option (1-5): 2

ID	Title	Author	Year	Copies	Status
1001	COMPUTER NETWORKS AND INTERNETS	DOUGLAS E. COMER	2015	10	Active
1002	IOS PROGRAMMING	CHRISTIAN KEUR	2016	10	Active
1003	AI-ASSISTED PROGRAMMING	TOM TAULLI	2024	12	Active
1004	INTRODUCTION TO PROGRAMMING WITH JAVA	JOHN DEAN	2013	20	Active
1005	MUST KNOW HIGH SCHOOL COMPUTER PROGRAMMING	JULIE SWAY	2020	15	Active
1006	DIGITAL MAVERICKS	DEBBIE SOON	2025	10	Active
1007	CONCEPTS OF PROGRAMMING LANGUAGES	SEBESTA	2015	12	Active
1008	BAYESIAN METHODS FOR HACKERS	CAMERON DAVIDSON-PILON	2015	20	Active
1009	C PROGRAMMING ABSOLUTE BEGINNER'S GUIDE	PERRY	2013	10	Active
1010	R PROGRAMMING FOR MASS SPECTROMETRY	RANDALL K. JULIAN	2025	18	Active
1011	DATABASE SYSTEMS	THOMAS CONNOLLY	2015	25	Active
1012	MODERN DATABASE MANAGEMENT	HOFFER	2016	15	Active
1013	PATRIOT	ALEXEI NAVALNY	2025	20	Active
1014	PLUCKED	MARYN MCKENNA	2019	10	Active
1015	COMPUTER SYSTEMS	RANDAL E. BRYANT	2016	15	Active
1016	ENGLISH LANGUAGE SYSTEMS	NARUTHAI	2025	20	Active
1017	DIGITAL SAT STUDY GUIDE	BRIAN W. STEWART	2025	16	Active
1018	BUSINESS PARTNER A1	MARGARET O'KEEFFE	2021	20	Active
1019	COMPUTER APPLICATION	JONATHAN CHAN	2015	15	Active
1020	SWIFT FOR PROGRAMMERS	PAUL DEITEL	2015	20	Active
1021	Test book	test	2025	5	Active

รูปภาพที่ 3- 14 แสดงข้อมูล View All Book

3.3.3 กรอกหมายเลข 2 เพื่อแสดงข้อมูลทั้งหมดของผู้ใช้งานที่มีในโปรแกรม



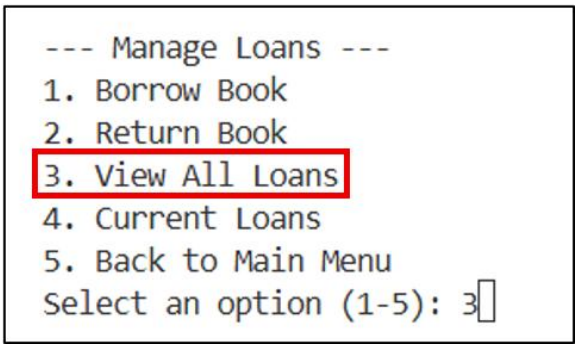
รูปภาพที่ 3- 15 การเลือกใช้งานฟังก์ชั่น View All Members

3.3.4 เมื่อกดเลือกหมายเลข 2 จะปรากฏข้อมูลหนังสือทั้งหมดดังภาพที่ 3-16

```
Select an option (1-5): 2
--------------------------------------------------------------------------------
ID        Name                Birth Year        Max Loan          Status
--------------------------------------------------------------------------------
68001     Daniel              2000              5                 Active
68002     Michael             2002              5                 Active
68003     Benjamin            2003              5                 Active
68004     Matthew             1999              5                 Active
68005     Sofia               2003              5                 Active
68006     Thomas              1995              5                 Active
68007     Henry               2005              5                 Active
68008     Steven              2001              5                 Active
68009     Olivia              2004              5                 Active
68010     Lucas               1996              5                 Active
68060     ROPEZ               2001              5                 Active
--------------------------------------------------------------------------------
```

รูปภาพที่ 3- 16 แสดงข้อมูล View All Members

3.3.5 กรอกหมายเลข 3 เพื่อแสดงข้อมูลทั้งหมดของการยืม - คืนหนังสือที่มีในโปรแกรม



รูปภาพที่ 3- 17 การเลือกใช้งานฟังก์ชั่น View All Loans

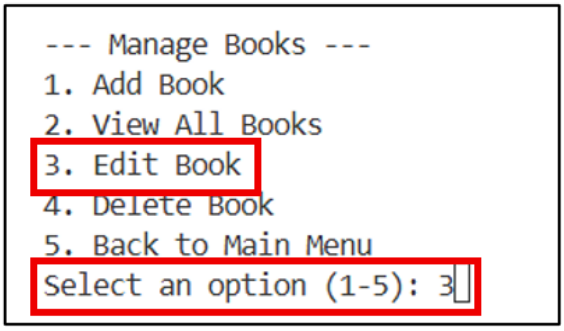
3.3.6 เมื่อกดเลือกหมายเลข 3 จะปรากฏข้อมูลการยืม - คืนหนังสือทั้งหมดดังภาพที่ 3-18

```
Select an option (1-5): 3
-------------------------------------------------------------------------------------------------------------------------
Timestamp           BookID  Title                                       MemberID   Member Name          Type     Status
-------------------------------------------------------------------------------------------------------------------------
2025-09-24 17:31:53  1005    MUST KNOW HIGH SCHOOL COMPUTER PROGRAMMING   68001      Daniel               Borrow   Borrowed
2025-09-24 17:32:12  1018    BUSINESS PARTNER A1                          68001      Daniel               Borrow   Borrowed
2025-09-24 17:32:33  1005    MUST KNOW HIGH SCHOOL COMPUTER PROGRAMMING   68001      Daniel               Borrow   Borrowed
2025-09-24 17:32:49  1007    CONCEPTS OF PROGRAMMING LANGUAGES            68001      Daniel               Borrow   Borrowed
2025-09-24 17:33:01  1016    ENGLISH LANGUAGE SYSTEMS                     68001      Daniel               Borrow   Borrowed
2025-09-24 17:34:44  1020    SWIFT FOR PROGRAMMERS                        68002      Michael              Borrow   Borrowed
2025-09-24 18:48:23  1007    CONCEPTS OF PROGRAMMING LANGUAGES            68001      Daniel               Return   Returned
2025-09-24 18:49:53  1003    AI-ASSISTED PROGRAMMING                      68008      Steven               Borrow   Borrowed
2025-09-24 18:50:04  1014    PLUCKED                                      68007      Henry                Borrow   Borrowed
2025-09-24 18:50:36  1011    DATABASE SYSTEMS                             68010      Lucas                Borrow   Borrowed
2025-09-24 18:50:52  1009    C PROGRAMMING ABSOLUTE BEGINNER'S GUIDE      68006      Thomas               Borrow   Borrowed
2025-09-24 18:51:55  1007    CONCEPTS OF PROGRAMMING LANGUAGES            68005      Sofia                Borrow   Borrowed
2025-09-24 18:52:16  1013    PATRIOT                                      68004      Matthew              Borrow   Borrowed
2025-09-24 18:52:44  1014    PLUCKED                                      68009      Olivia               Borrow   Borrowed
2025-09-24 18:53:40  1011    DATABASE SYSTEMS                             68009      Olivia               Borrow   Borrowed
2025-09-24 18:53:49  1011    DATABASE SYSTEMS                             68002      Michael              Borrow   Borrowed
2025-09-24 18:55:27  1006    DIGITAL MAVERICKS                            68005      Sofia                Borrow   Borrowed
2025-09-24 18:58:00  1001    COMPUTER NETWORKS AND INTERNETS              68010      Lucas                Borrow   Borrowed
2025-09-24 18:58:10  1001    COMPUTER NETWORKS AND INTERNETS              68004      Matthew              Borrow   Borrowed
-------------------------------------------------------------------------------------------------------------------------
```

รูปภาพที่ 3- 18 แสดงข้อมูล View All Loans

3.4 การใช้งานโปรแกรมแก้ไขข้อมูล

3.4.1 กรอกหมายเลข 3 เพื่อแก้ไขข้อมูลของหนังสือที่มีในโปรแกรม



รูปภาพที่ 3- 19 การเลือกใช้งานฟังก์ชั่น Edit All Book

3.4.2 เมื่อกดเลือกหมายเลข 3 จะปรากฏให้แก้ไขข้อมูลหนังสือของหัวข้อนั้นๆดังภาพที่ 3-20

```
Select an option (1-5): 3
------------------------------------------------------------------------------------------------------------
ID     Title                                        Author                    Year   Copies  Status
------------------------------------------------------------------------------------------------------------
1001   COMPUTER NETWORKS AND INTERNETS              DOUGLAS E. COMER          2015   10      Active
1002   IOS PROGRAMMING                              CHRISTIAN KEUR            2016   10      Active
1003   AI-ASSISTED PROGRAMMING                      TOM TAULLI                2024   12      Active
1004   INTRODUCTION TO PROGRAMMING WITH JAVA        JOHN DEAN                 2013   20      Active
1005   MUST KNOW HIGH SCHOOL COMPUTER PROGRAMMING   JULIE SWAY                2020   15      Active
1006   DIGITAL MAVERICKS                            DEBBIE SOON               2025   10      Active
1007   CONCEPTS OF PROGRAMMING LANGUAGES            SEBESTA                   2015   12      Active
1008   BAYESIAN METHODS FOR HACKERS                 CAMERON DAVIDSON-PILON    2015   20      Active
1009   C PROGRAMMING ABSOLUTE BEGINNER'S GUIDE      PERRY                     2013   10      Active
1010   R PROGRAMMING FOR MASS SPECTROMETRY          RANDALL K. JULIAN         2025   18      Active
1011   DATABASE SYSTEMS                             THOMAS CONNOLLY           2015   25      Active
1012   MODERN DATABASE MANAGEMENT                   HOFFER                    2016   15      Active
1013   PATRIOT                                      ALEXEI NAVALNY            2025   20      Active
1014   PLUCKED                                      MARYN MCKENNA             2019   10      Active
1015   COMPUTER SYSTEMS                             RANDAL E. BRYANT          2016   15      Active
1016   ENGLISH LANGUAGE SYSTEMS                     NARUTHAI                  2025   20      Active
1017   DIGITAL SAT STUDY GUIDE                      BRIAN W. STEWART          2025   16      Active
1018   BUSINESS PARTNER A1                          MARGARET O'KEEFFE         2021   20      Active
1019   COMPUTER APPLICATION                         JONATHAN CHAN             2015   15      Active
1020   SWIFT FOR PROGRAMMERS                        PAUL DEITEL               2015   20      Active
1021   COMPRO                                       David                     1900   5       Active
------------------------------------------------------------------------------------------------------------
Enter Book ID to edit: 1021
Editing Book ID 1021
Enter new Title [COMPRO]: COMPUTER PROGRAMING
Enter new Author [David]:
Enter new Year [1900]: 2019
Enter new Copies [5]:

✅ Book ID 1021 updated successfully
```

รูปภาพที่ 3- 20 การแก้ไขข้อมูล Edit All Book

3.4.3 กรอกหมายเลข 3 เพื่อแก้ไขข้อมูลของผู้ใช้งานที่มีในโปรแกรม

```
--- Manage Members ---
1. Add Member
2. View All Members
3. Edit Member
4. Delete Member
5. Back to Main Menu
Select an option (1-5): 3
```

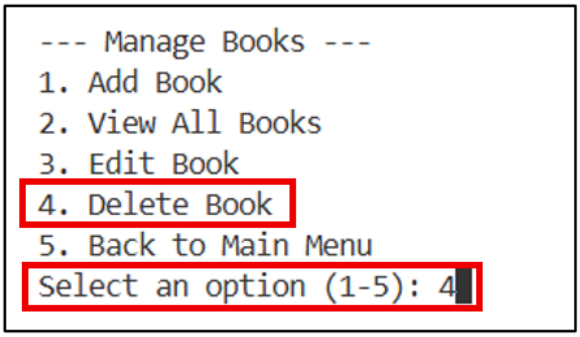
รูปภาพที่ 3- 21 การเลือกใช้งานฟังก์ชั่น Edit All Members

3.4.4 เมื่อกดเลือกหมายเลข 3 จะปรากฏให้แก้ไขข้อมูลหนังสือของหัวข้อนั้นๆดังภาพที่ 3-22

รูปภาพที่ 3- 22 การแก้ไขข้อมูล Edit All Members

3.5 การใช้งานโปรแกรมลบข้อมูล

3.5.1 กรอกหมายเลข 4 เพื่อลบหนังสือที่มีในโปรแกรม



รูปภาพที่ 3- 23 การเลือกใช้งานฟังก์ชั่น Delete Book

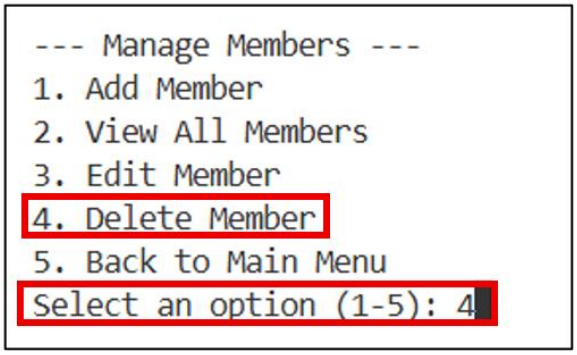
3.5.2 เมื่อกดเลือกหมายเลข 4 จะปรากฏการลบหนังสือเล่มนั้นๆออกจากระบบดังภาพที่ 3-23

```
Select an option (1-5): 4
------------------------------------------------------------------------------------------------------------
ID     Title                                        Author                    Year   Copies  Status
------------------------------------------------------------------------------------------------------------
1001   COMPUTER NETWORKS AND INTERNETS              DOUGLAS E. COMER          2015   10      Active
1002   IOS PROGRAMMING                              CHRISTIAN KEUR            2016   10      Active
1003   AI-ASSISTED PROGRAMMING                      TOM TAULLI                2024   12      Active
1004   INTRODUCTION TO PROGRAMMING WITH JAVA        JOHN DEAN                 2013   20      Active
1005   MUST KNOW HIGH SCHOOL COMPUTER PROGRAMMING   JULIE SWAY                2020   15      Active
1006   DIGITAL MAVERICKS                            DEBBIE SOON               2025   10      Active
1007   CONCEPTS OF PROGRAMMING LANGUAGES            SEBESTA                   2015   12      Active
1008   BAYESIAN METHODS FOR HACKERS                 CAMERON DAVIDSON-PILON    2015   20      Active
1009   C PROGRAMMING ABSOLUTE BEGINNER'S GUIDE      PERRY                     2013   10      Active
1010   R PROGRAMMING FOR MASS SPECTROMETRY          RANDALL K. JULIAN         2025   18      Active
1011   DATABASE SYSTEMS                             THOMAS CONNOLLY           2015   25      Active
1012   MODERN DATABASE MANAGEMENT                   HOFFER                    2016   15      Active
1013   PATRIOT                                      ALEXEI NAVALNY            2025   20      Active
1014   PLUCKED                                      MARYN MCKENNA             2019   10      Active
1015   COMPUTER SYSTEMS                             RANDAL E. BRYANT          2016   15      Active
1016   ENGLISH LANGUAGE SYSTEMS                     NARUTHAI                  2025   20      Active
1017   DIGITAL SAT STUDY GUIDE                      BRIAN W. STEWART          2025   16      Active
1018   BUSINESS PARTNER A1                          MARGARET O'KEEFFE         2021   20      Active
1019   COMPUTER APPLICATION                         JONATHAN CHAN             2015   15      Active
1020   SWIFT FOR PROGRAMMERS                        PAUL DEITEL               2015   20      Active
1021   COMPUTER PROGRAMING                          David                     2019   5       Active
------------------------------------------------------------------------------------------------------------
Enter Book ID: 1021

✅ Book ID 1021 deleted successfully
```

รูปภาพที่ 3- 24 การลบข้อมูล Delete Book

3.5.3 กรอกหมายเลข 4 เพื่อลบผู้ใช้งานที่มีในโปรแกรม



รูปภาพที่ 3- 25 การเลือกใช้งานฟังก์ชั่น Delete Members

3.5.4 เมื่อกดเลือกหมายเลข 4 จะปรากฏการลบผู้ใช้งานออกจากระบบดังภาพที่ 3-25

```
Select an option (1-5): 4
-------------------------------------------------------------------------------------
ID        Name                Birth Year        Max Loan          Status
-------------------------------------------------------------------------------------
68001     Daniel              2000              5                 Active
68002     Michael             2002              5                 Active
68003     Benjamin            2003              5                 Active
68004     Matthew             1999              5                 Active
68005     Sofia               2003              5                 Active
68006     Thomas              1995              5                 Active
68007     Henry               2005              5                 Active
68008     Steven              2001              5                 Active
68009     Olivia              2004              5                 Active
68010     Lucas               1995              5                 Active
68060     ROPEZ               2001              5                 Active
-------------------------------------------------------------------------------------
Enter Member ID: 68010

✅ Member ID 68010 deleted successfully
```

รูปภาพที่ 3- 26 การลบข้อมูล Delete Members

3.6 การใช้งานโปรแกรมยืม - คืนหนังสือ

3.6.1 กรอกหมายเลข 1 เพื่อยืมหนังสือที่มีในโปรแกรม

```
--- Manage Loans ---
1. Borrow Book
2. Return Book
3. View All Loans
4. Current Loans
5. Back to Main Menu
Select an option (1-5): 1
```

รูปภาพที่ 3- 27 การเลือกใช้งานฟังก์ชั่น Borrow Book

3.6.2 เมื่อกดเลือกหมายเลข 1 จะปรากฏการเลือกยืมหนังสือด้วยรหัสของหนังสือ และใช้รหัส Members ในการยืมหนังสือดังภาพที่ 3-27

```
Select an option (1-5): 1

=== Borrow Book ===
Available Books:
------------------------------------------------------------------------------------------
ID     Title                                          Copies   Borrowed   Available
------------------------------------------------------------------------------------------
1001   COMPUTER NETWORKS AND INTERNETS                10       2          8
1002   IOS PROGRAMMING                                10       0          10
1003   AI-ASSISTED PROGRAMMING                        12       1          11
1004   INTRODUCTION TO PROGRAMMING WITH JAVA          20       0          20
1005   MUST KNOW HIGH SCHOOL COMPUTER PROGRAMMING     15       1          14
1006   DIGITAL MAVERICKS                              10       1          9
1007   CONCEPTS OF PROGRAMMING LANGUAGES              12       1          11
1008   BAYESIAN METHODS FOR HACKERS                   20       0          20
1009   C PROGRAMMING ABSOLUTE BEGINNER'S GUIDE        10       1          9
1010   R PROGRAMMING FOR MASS SPECTROMETRY            18       0          18
1011   DATABASE SYSTEMS                               25       3          22
1012   MODERN DATABASE MANAGEMENT                     15       0          15
1013   PATRIOT                                        20       1          19
1014   PLUCKED                                        10       2          8
1015   COMPUTER SYSTEMS                               15       0          15
1016   ENGLISH LANGUAGE SYSTEMS                       20       1          19
1017   DIGITAL SAT STUDY GUIDE                        16       0          16
1018   BUSINESS PARTNER A1                            20       1          19
1019   COMPUTER APPLICATION                           15       0          15
1020   SWIFT FOR PROGRAMMERS                          20       1          19
------------------------------------------------------------------------------------------
Enter Book ID to borrow: 1010
Enter Member ID: 68009

✅ Member 'Olivia' borrowed 'R PROGRAMMING FOR MASS SPECTROMETRY' until 2025/10/29
```

รูปภาพที่ 3- 28 การยืมหนังสือ

3.6.3 กรอกหมายเลข 2 เพื่อคืนหนังสือในโปรแกรม

```
--- Manage Loans ---
1. Borrow Book
2. Return Book
3. View All Loans
4. Current Loans
5. Back to Main Menu
Select an option (1-5): 2
```

รูปภาพที่ 3- 29 การเลือกใช้งานฟังก์ชั่น Return Book

3.6.4 เมื่อกดเลือกหมายเลข 2 จะปรากฏหน้าการยืมหนังสือทั้งหมด ใส่ Book ID ที่ต้องการคืน ตามด้วย Member ID จะเป็นการคืนหนังสือดังภาพที่ 3-30

```
Select an option (1-5): 2

=== Return Book ===
------------------------------------------------------------------------------------------------
BookID   Title                                          MemberID   Member Name          Loan Date
------------------------------------------------------------------------------------------------
1005     MUST KNOW HIGH SCHOOL COMPUTER PROGRAMMING     68001      Daniel               2025/09/24
1018     BUSINESS PARTNER A1                            68001      Daniel               2025/09/24
1016     ENGLISH LANGUAGE SYSTEMS                       68001      Daniel               2025/09/24
1020     SWIFT FOR PROGRAMMERS                          68002      Michael              2025/09/24
1003     AI-ASSISTED PROGRAMMING                        68008      Steven               2025/09/24
1014     PLUCKED                                        68007      Henry                2025/09/24
1011     DATABASE SYSTEMS                               68010      Lucas                2025/09/24
1009     C PROGRAMMING ABSOLUTE BEGINNER'S GUIDE        68006      Thomas               2025/09/24
1007     CONCEPTS OF PROGRAMMING LANGUAGES              68005      Sofia                2025/09/24
1013     PATRIOT                                        68004      Matthew              2025/09/24
1014     PLUCKED                                        68009      Olivia               2025/09/24
1011     DATABASE SYSTEMS                               68009      Olivia               2025/09/24
1011     DATABASE SYSTEMS                               68002      Michael              2025/09/24
1006     DIGITAL MAVERICKS                              68005      Sofia                2025/09/24
1001     COMPUTER NETWORKS AND INTERNETS                68010      Lucas                2025/09/24
1001     COMPUTER NETWORKS AND INTERNETS                68004      Matthew              2025/09/24
1010     R PROGRAMMING FOR MASS SPECTROMETRY            68009      Olivia               2025/09/29
------------------------------------------------------------------------------------------------
Enter Book ID to return: 1010
Enter Member ID: 68009

✅ Book ID 1010 has been returned by Member ID 68009
```

รูปภาพที่ 3- 30 การคืนหนังสือ

3.6.5 กรอกหมายเลข 4 เพื่อแสดงข้อมูลหนังสือที่ถูกยืมในปัจจุบันในโปรแกรม

```
--- Manage Loans ---
1. Borrow Book
2. Return Book
3. View All Loans
4. Current Loans
5. Back to Main Menu
Select an option (1-5): 4
```

รูปภาพที่ 3- 31 การเลือกใช้งานฟังก์ชั่น Current Loans

3.6.6 เมื่อกดเลือกหมายเลข 4 จะแสดงการยืมหนังสือทั้งหมดดังภาพที่ 3-31

```
Select an option (1-5): 4

=== Current Loans ===
```

BookID	Title	MemberID	Member Name	Loan Date
1005	MUST KNOW HIGH SCHOOL COMPUTER PROGRAMMING	68001	Daniel	2025/09/24
1018	BUSINESS PARTNER A1	68001	Daniel	2025/09/24
1016	ENGLISH LANGUAGE SYSTEMS	68001	Daniel	2025/09/24
1020	SWIFT FOR PROGRAMMERS	68002	Michael	2025/09/24
1003	AI-ASSISTED PROGRAMMING	68008	Steven	2025/09/24
1014	PLUCKED	68007	Henry	2025/09/24
1011	DATABASE SYSTEMS	68010	Lucas	2025/09/24
1009	C PROGRAMMING ABSOLUTE BEGINNER'S GUIDE	68006	Thomas	2025/09/24
1007	CONCEPTS OF PROGRAMMING LANGUAGES	68005	Sofia	2025/09/24
1013	PATRIOT	68004	Matthew	2025/09/24
1014	PLUCKED	68009	Olivia	2025/09/24
1011	DATABASE SYSTEMS	68009	Olivia	2025/09/24
1011	DATABASE SYSTEMS	68002	Michael	2025/09/24
1006	DIGITAL MAVERICKS	68005	Sofia	2025/09/24
1001	COMPUTER NETWORKS AND INTERNETS	68010	Lucas	2025/09/24
1001	COMPUTER NETWORKS AND INTERNETS	68004	Matthew	2025/09/24

รูปภาพที่ 3- 32 แสดงการยืมหนังสือ

บทที่ 4
อธิบายการทำงานของ Code

4.1 ฟังก์ชั่นไบนารีพื้นฐานในระบบยืม – คืนหนังสือห้องสมุด

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

```
import struct
```

รูปภาพที่ 4- 1 Code Module pickle

4.1.2 import time คือคำสั่งในภาษา Python ที่ใช้สำหรับนำเข้า (import) โมดูล time มาใช้งานโมดูล time จะช่วยให้เราสามารถทำงานที่เกี่ยวข้องกับเวลา เช่น หน่วงเวลา (delay) จับเวลาการทำงาน แปลงเวลาให้อ่านง่าย import time จะเรียกใช้โมดูลที่เกี่ยวกับ เวลา เพื่อหยุดเวลา,จับเวลา, หรือจัดการเวลา ได้สะดวกขึ้น

```
import time
```

รูปภาพที่ 4- 2 Code Module time

4.1.3 datetime เป็น class หลัก ที่รวม วันและเวลา เข้าด้วยกัน (ต่างจาก date ที่มีแค่วัน หรือ time ที่มีแค่เวลา)

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 book struck "<i50si30siiil" คือ format string กำหนด layout ของ record แต่ละเล่มดังรูปที่ 4-3

```
books_struck = struct.Struct("<i50si30siiiI")
```

รูปภาพที่ 4- 3 โครงสร้างข้อมูล book struck

4.1.7 ฟังก์ชั่น add_book

ฟังก์ชัน add_book มีหน้าที่สร้างข้อมูลหนังสือ 1 เล่มแล้วบันทึกลงไฟล์ books.dat ในรูปแบบไบนารี โดยเริ่มจากการเก็บค่าเวลาปัจจุบัน (now) ในรูปแบบ Unix timestamp เพื่อนำไปใช้เป็นเวลาสร้างและเวลาปรับปรุงล่าสุด จากนั้นนำข้อมูลต่าง ๆ ได้แก่ รหัสหนังสือ (book_id), ชื่อหนังสือ (title), สถานะ (status), ผู้แต่ง (author), ปีพิมพ์ (year), จำนวนเล่ม (copies), และเวลา (created_at, updated_at) มาจัดรูปแบบตามโครงสร้างที่กำหนดใน books_struck.pack โดยจะมีการแปลงข้อความเป็น bytes และปรับความยาวให้คงที่ด้วยการเติมค่า \x00 เพื่อให้แต่ละเรคอร์ดมีขนาดเท่ากันเสมอ สุดท้ายเปิดไฟล์ books.dat ในโหมด append binary ("ab") แล้วเขียนข้อมูลเรคอร์ดที่สร้างขึ้นต่อท้ายไฟล์ ทำให้สามารถเก็บข้อมูลหนังสือเพิ่มได้ทีละเล่มโดยไม่ไปทับข้อมูลเดิมที่มีอยู่ก่อนหน้า

```
def add_book(book_id, title, status, author, year, copies):
    now = int(time.time())
    record = books_struck.pack(
        book_id,
        title.encode("utf-8").ljust(50, b"\x00"),
        status,
        author.encode("utf-8").ljust(30, b"\x00"),
        year,
        copies,
        now,    # created_at
        now     # updated_at
    )
    with open("books.dat", "ab") as f:
        f.write(record)

members_struck = struct.Struct("<ii50siiII")
```

รูปภาพที่ 4- 4 ฟังก์ชั่น add_book

4.1.8 โครงสร้างข้อมูล members_struck

"<ii50siiII" คือ โครงสร้างข้อมูล (record format) สำหรับสมาชิก (members) โดยใช้ struct.Struct ดังรูปที่ 4-5

```
members_struck = struct.Struct("<ii50siiII")
```

รูปภาพที่ 4- 5 โครงสร้างข้อมูล members struck

4.1.9 ฟังก์ชั่น add_members

ฟังก์ชัน add_members มีหน้าที่บันทึกข้อมูลสมาชิกใหม่ลงไฟล์ members.dat โดยใช้โครงสร้างที่กำหนดใน members_struck เพื่อให้ข้อมูลแต่ละเรคอร์ดมีขนาดคงที่ เริ่มต้นด้วยการเก็บเวลาปัจจุบัน (now) ในรูปแบบ Unix timestamp สำหรับใช้เป็นค่า created_at และ updated_at จากนั้นนำข้อมูลที่รับเข้ามา ได้แก่ รหัสสมาชิก (member_id), สถานะ (status), ชื่อ (name), ปีเกิด (birth_year) และจำนวนยืมสูงสุด (max_loan) มาจัดรูปแบบให้ตรงกับโครงสร้าง โดยชื่อจะถูกแปลงเป็นข้อมูลแบบ bytes และบังคับความยาว 50 ไบต์ด้วยการเติม \x00 จากนั้นข้อมูลทั้งหมดถูก pack เป็นเรคอร์ดไบนารี ก่อนจะเปิดไฟล์ members.dat ในโหมด append binary ("ab") และเขียนข้อมูลเรคอร์ดลงไปท้ายไฟล์ ทำให้สามารถเพิ่มสมาชิกใหม่ได้ต่อเนื่องโดยไม่กระทบกับข้อมูลเดิมที่มีอยู่แล้ว.

```
def add_members(member_id , status , name, birth_year , max_loan):
    now = int(time.time())
    record = members_struck.pack(
        member_id,
        status,
        name.encode("utf-8").ljust(50, b"\x00"),
        birth_year,
        max_loan,
        now,    # created_at
        now     # updated_at
    )
    with open("members.dat", "ab") as f:
        f.write(record)
```

รูปภาพที่ 4- 6 ฟังก์ชั่น add_members

4.1.10 โครงสร้างข้อมูล loans_struck

"<Iiii10s10sii" คือ เป็นการสร้าง โครงสร้างข้อมูล (binary record format) สำหรับเก็บข้อมูลการยืมหนังสือ (loans) โดยใช้ struct.Struct ดังรูปที่ 4-7

```
loans_struck = struct.Struct("<Iiii10s10sii")
```

รูปภาพที่ 4- 7 โครงสร้างข้อมูล loans struck

4.1.11 ฟังก์ชั่น add_loans

ฟังก์ชัน add_loans ถูกออกแบบมาเพื่อบันทึกข้อมูลการยืม–คืนหนังสือแต่ละครั้งลงไฟล์ loans.dat โดยเก็บข้อมูลเป็นเรคอร์ดแบบไบนารีตามโครงสร้างที่กำหนดใน loans_struck ภายในฟังก์ชันจะเริ่มจากการเก็บเวลาปัจจุบัน (now) ในรูปแบบ Unix timestamp เพื่อใช้เป็นค่าไอดีหรือหมายเลขอ้างอิงของการทำรายการ (op_code จะเป็นรหัสการดำเนินการ เช่น ยืมหรือคืน) จากนั้นจะรวมข้อมูลที่รับเข้ามา ได้แก่ รหัสหนังสือ (book_id), รหัสสมาชิก (member_id), วันที่ยืม (loan_date), วันที่คืน (return_date), สถานะหลังทำรายการ (status_after) และค่าบอกว่ายืมอยู่หรือไม่ (is_rented_after) โดยวันที่จะถูกแปลงเป็น bytes และปรับความยาวคงที่ 10 ตัวอักษรด้วยการเติม \x00 เพื่อให้ตรงตามโครงสร้างที่กำหนด เมื่อข้อมูลทั้งหมดถูก pack เป็นเรคอร์ดแล้ว จะถูกเขียนต่อท้ายไฟล์ loans.dat ในโหมด append binary ("ab") ทำให้ไฟล์นี้สามารถเก็บประวัติการยืมและคืนหนังสือได้อย่างต่อเนื่องโดยไม่กระทบกับข้อมูลเดิมที่ถูกบันทึกไว้ก่อนหน้า

```python
def add_loans(op_code, book_id, member_id, loan_date, return_date, status_after, is_rented_after):
    now = int(time.time())
    record = loans_struck.pack(
        now,
        op_code,
        book_id,
        member_id,
        loan_date.encode("utf-8").ljust(10, b"\x00"),
        return_date.encode("utf-8").ljust(10, b"\x00"),
        status_after,
        is_rented_after
    )
    with open("loans.dat", "ab") as f:
        f.write(record)
```

รูปภาพที่ 4- 8 ฟังก์ชั่น add_loans

4.1.12 ฟังก์ชั่น read_all_books

ฟังก์ชัน read_all_books มีหน้าที่อ่านข้อมูลหนังสือทั้งหมดจากไฟล์ books.dat ในรูปแบบไบนารีแล้วคืนค่าเป็นรายการของ dictionary แต่ละรายการแทนหนังสือหนึ่งเล่ม โดยเริ่มจากสร้างลิสต์ว่าง books และพยายามเปิดไฟล์ในโหมดอ่านไบนารี ("rb") หากไฟล์ไม่พบจะจับ FileNotFoundError แล้วพิมพ์ข้อความแจ้งเตือน จากนั้นจะอ่านข้อมูลทีละเรคอร์ดตามขนาดที่กำหนดใน books_struck.size และหยุดเมื่ออ่านไม่ครบขนาดของเรคอร์ดแต่ละชิ้น ข้อมูลแต่ละเรคอร์ดจะถูก unpack เพื่อนำค่าต่าง ๆ เช่น รหัสหนังสือ (book_id), ชื่อหนังสือ (title), สถานะ (status), ผู้แต่ง (author), ปีพิมพ์ (year), จำนวนเล่ม (copies), เวลาสร้าง (created_at) และเวลาปรับปรุงล่าสุด (updated_at) มาแปลงเป็นรูปแบบที่อ่านง่าย โดยชื่อและผู้แต่งจะถูกแปลงจาก bytes เป็น string และตัด padding \x00 ส่วนเวลาจะถูกแปลงจาก Unix timestamp เป็นสตริงรูปแบบ YYYY-MM-DD HH:MM:SS ก่อนจะเก็บลงใน dictionary แล้วเพิ่มเข้าไปในลิสต์ books สุดท้ายฟังก์ชันจะคืนลิสต์ของหนังสือทั้งหมดที่อ่านได้

```python
def read_all_books(filename="books.dat"):
    books = []
    try:
        with open(filename, "rb") as f:
            while True:
                data = f.read(books_struck.size)
                if len(data) < books_struck.size:
                    break
                unpacked = books_struck.unpack(data)

                book = {
                    "book_id": unpacked[0],
                    "title": unpacked[1].decode("utf-8").rstrip("\x00"),
                    "status": unpacked[2],
                    "author": unpacked[3].decode("utf-8").rstrip("\x00"),
                    "year": unpacked[4],
                    "copies": unpacked[5],
                    "created_at": datetime.fromtimestamp(unpacked[6]).strftime("%Y-%m-%d %H:%M:%S"),
                    "updated_at": datetime.fromtimestamp(unpacked[7]).strftime("%Y-%m-%d %H:%M:%S")
                }
                books.append(book)
    except FileNotFoundError:
        print(f"ไฟล์ {filename} ไม่พบ")
    return books
```

รูปภาพที่ 4- 9 ฟังก์ชั่น read_all_books

4.1.13 ฟังก์ชั่น read_all_members

ฟังก์ชั่น read_all_members มีหน้าที่อ่านข้อมูลสมาชิกทั้งหมดจากไฟล์ members.dat ซึ่งเก็บเป็นไบนารี แล้วคืนค่าเป็นลิสต์ของ dictionary แต่ละตัวแทนสมาชิกหนึ่งคน โดยเริ่มจากสร้างลิสต์ว่าง members จากนั้นพยายามเปิดไฟล์ในโหมดอ่านไบนารี ("rb") หากไฟล์ไม่พบจะจับข้อผิดพลาด FileNotFoundError และพิมพ์ข้อความแจ้งเตือน สำหรับการอ่านข้อมูล ฟังก์ชันจะวนลูปอ่านทีละเรคอร์ดตามขนาดที่กำหนดใน members_struck.size และหยุดเมื่ออ่านไม่ครบขนาดของเรคอร์ด ข้อมูลแต่ละเรคอร์ดจะถูก unpack เพื่อนำค่าต่าง ๆ เช่น รหัสสมาชิก (member_id), สถานะ (status), ชื่อ (name), ปีเกิด (birth_year), จำนวนครั้งที่สามารถยืมสูงสุด (max_loan), เวลาสร้าง (created_at) และเวลาปรับปรุงล่าสุด (updated_at) มาจัดเก็บ โดยชื่อจะถูกแปลงจาก bytes เป็น string และตัด padding \x00 ส่วนเวลาจะถูกแปลงจาก Unix timestamp เป็นสตริงรูปแบบ YYYY-MM-DD HH:MM:SS ก่อนจะเพิ่ม dictionary ของสมาชิกแต่ละคนลงในลิสต์ members สุดท้ายฟังก์ชันคืนลิสต์สมาชิกทั้งหมดที่อ่านได้

```python
def read_all_members(filename="members.dat"):
    members = []
    try:
        with open(filename, "rb") as f:
            while True:
                data = f.read(members_struck.size)
                if len(data) < members_struck.size:
                    break
                unpacked = members_struck.unpack(data)

                member = {
                    "member_id": unpacked[0],
                    "status": unpacked[1],
                    "name": unpacked[2].decode("utf-8").rstrip("\x00"),
                    "birth_year": unpacked[3],
                    "max_loan": unpacked[4],
                    "created_at": datetime.fromtimestamp(unpacked[5]).strftime("%Y-%m-%d %H:%M:%S"),
                    "updated_at": datetime.fromtimestamp(unpacked[6]).strftime("%Y-%m-%d %H:%M:%S")
                }
                members.append(member)
    except FileNotFoundError:
        print(f"ไฟล์ {filename} ไม่พบ")
    return members
```

รูปภาพที่ 4- 10 ฟังก์ชั่น read_all_members

4.1.14 ฟังก์ชั่น read_all_loans

ฟังก์ชัน read_all_loans มีหน้าที่อ่านข้อมูลการยืม–คืนหนังสือทั้งหมดจากไฟล์ loans.dat ซึ่งเก็บเป็นไบนารี แล้วคืนค่าเป็นลิสต์ของ dictionary แต่ละตัวแทนรายการยืมหนึ่งรายการ โดยเริ่มจากสร้างลิสต์ว่าง loans จากนั้นพยายามเปิดไฟล์ในโหมดอ่านไบนารี ("rb") หากไฟล์ไม่พบจะจับข้อผิดพลาด FileNotFoundError และพิมพ์ข้อความแจ้งเตือน สำหรับการอ่านข้อมูล ฟังก์ชันจะวนลูปอ่านทีละเรคอร์ดตามขนาดที่กำหนดใน loans_struck.size และหยุดเมื่ออ่านไม่ครบขนาดของเรคอร์ด ข้อมูลแต่ละเรคอร์ดจะถูก unpack เพื่อนำค่าต่าง ๆ เช่น เวลาของรายการ (ts), รหัสการดำเนินการ (op_code), รหัสหนังสือ (book_id), รหัสสมาชิก (member_id), วันที่ยืม (loan_date), วันที่คืน (return_date), สถานะหลังทำรายการ (status_after) และค่าบอกว่ายืมอยู่หรือไม่ (is_rented_after) มาแปลงเป็นรูปแบบที่อ่านง่าย โดยวันที่ยืมและคืนจะถูกแปลงจาก bytes เป็น string และตัด padding \x00 ส่วน timestamp จะถูกแปลงเป็นสตริงรูปแบบ YYYY-MM-DD HH:MM:SS ก่อนจะเพิ่ม dictionary ของรายการยืมแต่ละรายการลงในลิสต์ loans สุดท้ายฟังก์ชันคืนลิสต์รายการยืมทั้งหมดที่อ่านได้

```python
def read_all_loans(filename="loans.dat"):
    loans = []
    try:
        with open(filename, "rb") as f:
            while True:
                data = f.read(loans_struck.size)
                if len(data) < loans_struck.size:
                    break
                unpacked = loans_struck.unpack(data)
                loan = {
                    "ts": datetime.fromtimestamp(unpacked[0]).strftime("%Y-%m-%d %H:%M:%S"),
                    "op_code": unpacked[1],
                    "book_id": unpacked[2],
                    "member_id": unpacked[3],
                    "loan_date": unpacked[4].decode("utf-8").rstrip("\x00"),
                    "return_date": unpacked[5].decode("utf-8").rstrip("\x00"),
                    "status_after": unpacked[6],
                    "is_rented_after": unpacked[7],
                }
                loans.append(loan)
    except FileNotFoundError:
        print(f"\n❌ File {filename} not found")
    return loans
```

รูปภาพที่ 4- 11 ฟังก์ชั่น read_all_loans

4.1.15 ฟังก์ชั่น read_all_loans

ฟังก์ชัน read_all_loans มีหน้าที่อ่านข้อมูลการยืม–คืนหนังสือทั้งหมดจากไฟล์ loans.dat ซึ่งเก็บเป็นไบนารี แล้วคืนค่าเป็นลิสต์ของ dictionary แต่ละตัวแทนรายการยืมหนึ่งรายการ โดยเริ่มจากสร้างลิสต์ว่าง loans จากนั้นพยายามเปิดไฟล์ในโหมดอ่านไบนารี ("rb") หากไฟล์ไม่พบจะจับข้อผิดพลาด FileNotFoundError และพิมพ์ข้อความแจ้งเตือน สำหรับการอ่านข้อมูล ฟังก์ชันจะวนลูปอ่านทีละเรคอร์ดตามขนาดที่กำหนดใน loans_struck.size และหยุดเมื่ออ่านไม่ครบขนาดของเรคอร์ด ข้อมูลแต่ละเรคอร์ดจะถูก unpack เพื่อนำค่าต่าง ๆ เช่น เวลาของรายการ (ts), รหัสการดำเนินการ (op_code), รหัสหนังสือ (book_id), รหัสสมาชิก (member_id), วันที่ยืม (loan_date), วันที่คืน (return_date), สถานะหลังทำรายการ (status_after) และค่าบอกว่ายืมอยู่หรือไม่ (is_rented_after) มาแปลงเป็นรูปแบบที่อ่านง่าย โดยวันที่ยืมและคืนจะถูกแปลงจาก bytes เป็น string และตัด padding \x00 ส่วน timestamp จะถูกแปลงเป็นสตริงรูปแบบ YYYY-MM-DD HH:MM:SS ก่อนจะเพิ่ม dictionary ของรายการยืมแต่ละรายการลงในลิสต์ loans สุดท้ายฟังก์ชันคืนลิสต์รายการยืมทั้งหมดที่อ่านได้

4.2 ฟังก์ชั่นเมนูระบบยืม – คืนหนังสือห้องสมุด

4.2.1 ฟังก์ชั่น menu_add_book

ฟังก์ชัน menu_add_book เป็น เมนูอินเตอร์เฟสแบบง่าย สำหรับเพิ่มหนังสือใหม่เข้าสู่ระบบ โดยเริ่มจากพิมพ์หัวข้อ "=== Add New Book ===" เพื่อบอกผู้ใช้ว่าอยู่ในหน้าการเพิ่มหนังสือ จากนั้นใช้ input() รับค่าต่าง ๆ ได้แก่ รหัสหนังสือ (book_id), ชื่อหนังสือ (title), ผู้แต่ง (author), ปีพิมพ์ (year) และจำนวนเล่ม (copies) พร้อมตั้งค่าเริ่มต้นของสถานะ (status = 1) ซึ่งอาจหมายถึงหนังสือพร้อมใช้งาน หลังจากรับข้อมูลครบ ฟังก์ชันจะเรียก add_book() เพื่อบันทึกหนังสือใหม่ลงไฟล์ books.dat และพิมพ์ข้อความยืนยันความสำเร็จ หากผู้ใช้ป้อนค่าที่ไม่ถูกต้อง (เช่น ตัวอักษรในช่องที่ควรเป็นตัวเลข) จะเกิด ValueError และฟังก์ชันจะแจ้งข้อความเตือนให้ป้อนข้อมูลใหม่อย่างถูกต้อง ทำให้ฟังก์ชันนี้ช่วยให้ผู้ใช้สามารถเพิ่มหนังสือใหม่ได้ง่ายและปลอดภัยต่อการป้อนข้อมูลผิดพลาด

```
def menu_add_book():
    print("\n=== Add New Book ===")
    try:
        book_id = int(input("Enter Book ID: "))
        title = str(input("Enter Title: "))
        author = str(input("Enter Author: "))
        year = int(input("Enter Year: "))
        copies = int(input("Enter Copies: "))
        status = 1

        add_book(book_id, title, status, author, year, copies)
        print(f"\n✅ Book '{title}' added successfully!")

    except ValueError:
        print("\n❌ Invalid input. Please enter valid information.")
```

รูปภาพที่ 4- 12 ฟังก์ชั่น menu_add_book

4.2.2 ฟังก์ชั่น menu_delete_book

ฟังก์ชัน menu_delete_book ใช้สำหรับลบหนังสือจากไฟล์ books.dat โดยจะอ่านข้อมูลหนังสือทั้งหมดเข้ามาในลิสต์ก่อน จากนั้นค้นหาหนังสือที่ตรงกับรหัส book_id และมีสถานะยังใช้งานอยู่ ถ้าพบจะเปลี่ยนสถานะเป็น 0 (ปิดการใช้งาน) และปรับค่า updated_at เป็นเวลาปัจจุบัน จากนั้นเขียนข้อมูลทั้งหมดกลับลงไฟล์แทนของเดิม หากไม่พบหนังสือหรือถูกลบไปแล้วจะแจ้งผู้ใช้ ฟังก์ชันนี้เป็นการลบแบบ soft delete ทำให้ข้อมูลยังอยู่ในไฟล์แต่ระบบจะถือว่าหนังสือเล่มนั้นถูกลบเรียบร้อยแล้ว

```python
def menu_delete_book(book_id, filename="books.dat"):
    books = []
    found = False
    try:
        with open(filename, "rb") as f:
            while True:
                data = f.read(books_struck.size)
                if len(data) < books_struck.size:
                    break
                unpacked = books_struck.unpack(data)
                books.append(list(unpacked))
    except FileNotFoundError:
        print(f"\n❌ File {filename} not found")
        return

    for b in books:
        if b[0] == book_id and b[2] == 1:
            b[2] = 0
            b[7] = int(time.time())
            found = True
            break

    if not found:
        print(f"\n❌ Book ID {book_id} not found or already deleted")
        return

    with open(filename, "wb") as f:
        for b in books:
            record = books_struck.pack(*b)
            f.write(record)

    print(f"\n✅ Book ID {book_id} deleted successfully")
```

รูปภาพที่ 4- 13 ฟังก์ชั่น menu_delete_book

4.2.3 ฟังก์ชั่น menu_view_books

ฟังก์ชัน menu_view_books ใช้สำหรับแสดงรายการหนังสือทั้งหมดจากไฟล์ books.dat โดยเริ่มจากเรียกฟังก์ชัน read_all_books เพื่อดึงข้อมูลหนังสือทั้งหมดเข้ามาเป็นลิสต์ หากไม่พบหนังสือจะแจ้งข้อความ "No books found." แล้วออกจากฟังก์ชัน จากนั้นพิมพ์หัวตารางและเส้นคั่นเพื่อจัดรูปแบบให้สวยงาม สำหรับแต่ละหนังสือในลิสต์จะแปลงสถานะเป็นข้อความ "Active" หากสถานะเป็น 1 หรือ "Deleted" หากสถานะเป็น 0 แล้วพิมพ์ข้อมูลหนังสือแต่ละรายการ เช่น รหัสหนังสือ, ชื่อ, ผู้แต่ง, ปีพิมพ์, จำนวนเล่ม และสถานะ โดยจัดเรียงให้อ่านง่ายด้วยการจัดความกว้างคอลัมน์ สุดท้ายพิมพ์เส้นคั่นเพื่อปิดท้ายตาราง ทำให้ผู้ใช้สามารถมองเห็นรายการหนังสือทั้งหมดพร้อมสถานะได้อย่างชัดเจน

```python
def menu_view_books(filename="books.dat"):
    books = read_all_books(filename)
    if not books:
        print("No books found.")
        return

    print("-" * 108)
    print(f"{'ID':<6} {'Title':<45} {'Author':<25} {'Year':<6} {'Copies':<7} {'Status':<8}")
    print("-" * 108)

    for b in books:
        status_text = "Active" if b['status'] == 1 else "Deleted"
        print(f"{b['book_id']:<6} {b['title']:<45} {b['author']:<25} {b['year']:<6} {b['copies']:<7} {status_text:<8}")

    print("-" * 108)
```

รูปภาพที่ 4- 14 ฟังก์ชั่น menu_view_books

4.2.4 ฟังก์ชั่น menu_edit_book

ฟังก์ชัน menu_edit_book ใช้สำหรับแก้ไขข้อมูลหนังสือในไฟล์ books.dat โดยเริ่มจากเรียกฟังก์ชัน menu_view_books เพื่อแสดงรายการหนังสือทั้งหมดให้ผู้ใช้เลือก จากนั้นรับรหัสหนังสือ (book_id) ที่ต้องการแก้ไข หากผู้ใช้ป้อนค่าไม่ถูกต้องจะหยุดฟังก์ชันและแจ้งเตือน หลังจากนั้นอ่านข้อมูลหนังสือทั้งหมดเข้าลิสต์ books ในรูปแบบไบนารีและแปลงเป็น list ของแต่ละเรคอร์ด ถัดไปจะค้นหาหนังสือที่ตรงกับรหัสและมีสถานะ Active (1) หากพบ จะให้ผู้ใช้ป้อนข้อมูลใหม่ เช่น ชื่อหนังสือ (title), ผู้แต่ง (author), ปีพิมพ์ (year) และจำนวนเล่ม (copies) โดยสามารถเว้นว่างเพื่อคงค่าของเดิมได้ หลังจากรับข้อมูลแล้ว จะอัปเดตค่าต่าง ๆ ในเรคอร์ดรวมถึงปรับ updated_at เป็นเวลาปัจจุบัน เมื่อแก้ไขเสร็จแล้ว เขียนข้อมูลทั้งหมดกลับลงไฟล์ในโหมดเขียนทับ ("wb") และแจ้งผู้ใช้ว่าการแก้ไขสำเร็จ หากไม่พบหนังสือหรือไม่ใช่ Active จะแจ้งข้อความว่าไม่พบหรือไม่สามารถแก้ไขได้ ฟังก์ชันนี้ช่วยให้ผู้ใช้สามารถปรับปรุงข้อมูลหนังสือได้อย่างสะดวกและปลอดภัย

```python
def menu_edit_book(filename="books.dat"):
    menu_view_books()
    try:
        book_id = int(input("Enter Book ID to edit: "))
    except ValueError:
        print("\n❌ Invalid input. Please enter a number.")
        return
    books = []
    found = False
    try:
        with open(filename, "rb") as f:
            while True:
                data = f.read(books_struck.size)
                if len(data) < books_struck.size:
                    break
                unpacked = books_struck.unpack(data)
                books.append(list(unpacked))
    except FileNotFoundError:
        print(f"\n❌ File {filename} not found")
        return
    for b in books:
        if b[0] == book_id and b[2] == 1:
            print(f"Editing Book ID {book_id}")
            title = input(f"Enter new Title [{b[1].decode('utf-8').rstrip(chr(0))}]: ")
            author = input(f"Enter new Author [{b[3].decode('utf-8').rstrip(chr(0))}]: ")
            try:
                year = input(f"Enter new Year [{b[4]}]: ")
                year = int(year) if year else b[4]
                copies = input(f"Enter new Copies [{b[5]}]: ")
                copies = int(copies) if copies else b[5]
            except ValueError:
                print("\n❌ Invalid number input. Edit canceled.")
                return
            b[1] = title.encode("utf-8").ljust(50, b"\x00") if title else b[1]
            b[3] = author.encode("utf-8").ljust(30, b"\x00") if author else b[3]
            b[4] = year
            b[5] = copies
            b[7] = int(time.time())
            found = True
            break
    if not found:
        print(f"\n❌ Book ID {book_id} not found or not active")
        return
    with open(filename, "wb") as f:
        for b in books:
            record = books_struck.pack(*b)
            f.write(record)
    print(f"\n✅ Book ID {book_id} updated successfully")
```

รูปภาพที่ 4- 15 ฟังก์ชั่น menu_edit_book

4.2.5 ฟังก์ชั่น menu_add_member

ฟังก์ชัน menu_add_member เป็นเมนูสำหรับเพิ่มสมาชิกใหม่เข้าสู่ระบบ โดยเริ่มจากพิมพ์หัวข้อ "=== Add New Member ===" เพื่อให้ผู้ใช้ทราบว่ากำลังอยู่ในหน้าการเพิ่มสมาชิก จากนั้นรับค่าต่าง ๆ ผ่าน input() ได้แก่ รหัสสมาชิก (member_id), ชื่อสมาชิก (name) และปีเกิด (birth_year) พร้อมตั้งค่าเริ่มต้นของสถานะ (status = 1) และจำนวนครั้งที่สามารถยืมสูงสุด (max_loan = 5) หลังจากรับข้อมูลครบ ฟังก์ชันจะเรียก add_members() เพื่อบันทึกข้อมูลสมาชิกลงไฟล์ members.dat และพิมพ์ข้อความยืนยันความสำเร็จ หากผู้ใช้ป้อนค่าที่ไม่ถูกต้อง เช่น ใส่ตัวอักษรในช่องที่ควรเป็นตัวเลข จะเกิด ValueError และฟังก์ชันจะแจ้งเตือนให้ป้อนข้อมูลใหม่อย่างถูกต้อง ทำให้การเพิ่มสมาชิกใหม่ปลอดภัยและสะดวก

```python
def menu_add_member():
    print("\n=== Add New Member ===")
    try:
        member_id = int(input("Enter Member ID: "))
        status = 1
        name = str(input("Enter Member Name: "))
        birth_year = int(input("Enter Birth Year: "))
        max_loan = 5

        add_members(member_id, status, name, birth_year, max_loan)
        print(f"\n✅ Member '{name}' added successfully!")

    except ValueError:
        print("\n❌ Invalid input. Please enter valid information.")
```

รูปภาพที่ 4- 16 ฟังก์ชั่น menu_add_member

4.2.6 ฟังก์ชั่น menu_view_members

ฟังก์ชัน menu_view_members ใช้สำหรับแสดงรายการสมาชิกทั้งหมดจากไฟล์ members.dat โดยเริ่มจากเรียก read_all_members เพื่อดึงข้อมูลสมาชิกเข้ามาเป็นลิสต์ หากไม่พบสมาชิกจะแจ้งข้อความ "No members found." แล้วออกจากฟังก์ชัน จากนั้นพิมพ์หัวตารางและเส้นคั่นเพื่อจัดรูปแบบให้สวยงาม สำหรับแต่ละสมาชิกในลิสต์ จะตรวจสอบสถานะ หาก status เป็น 1 จะแสดงเป็น "Active" หากเป็น 0 จะแสดงเป็น "Deleted" แล้วพิมพ์ข้อมูลสมาชิก เช่น รหัสสมาชิก, ชื่อ, ปีเกิด, จำนวนครั้งที่สามารถยืมสูงสุด และสถานะ โดยจัดความกว้างคอลัมน์ให้อ่านง่าย สุดท้ายพิมพ์เส้นคั่นเพื่อปิดท้ายตาราง ทำให้ผู้ใช้สามารถเห็นข้อมูลสมาชิกทั้งหมดพร้อมสถานะได้อย่างชัดเจน

```python
def menu_view_members(filename="members.dat"):
    members = read_all_members(filename)
    if not members:
        print("No members found.")
        return

    print("-" * 83)
    print(f"{'ID':<10} {'Name':<22} {'Birth Year':<17} {'Max Loan':<19} {'Status':<17}")
    print("-" * 83)

    for m in members:
        status_text = "Active" if m['status'] == 1 else "Deleted"
        print(f"{m['member_id']:<10} {m['name']:<22} {m['birth_year']:<17} {m['max_loan']:<19} {status_text:<17}")

    print("-" * 83)
```

รูปภาพที่ 4- 17 ฟังก์ชั่น menu_view_members

4.2.7 ฟังก์ชั่น menu_edit_member

ฟังก์ชัน menu_edit_member ใช้สำหรับแก้ไขข้อมูลสมาชิกในไฟล์ members.dat โดยเริ่มจากเรียก menu_view_members เพื่อแสดงรายการสมาชิกทั้งหมดให้ผู้ใช้เลือก จากนั้นรับรหัสสมาชิก (member_id) ที่ต้องการแก้ไข หากผู้ใช้ป้อนค่าที่ไม่ถูกต้องจะหยุดฟังก์ชันและแจ้งเตือน หลังจากนั้นอ่านข้อมูลสมาชิกทั้งหมดเข้าลิสต์ members ในรูปแบบไบนารีและแปลงเป็น list ของแต่ละเรคอร์ด ถัดไปจะค้นหาสมาชิกที่ตรงกับรหัสและมีสถานะ Active (1) หากพบ จะให้ผู้ใช้ป้อนข้อมูล

ใหม่ เช่น ชื่อ (name), ปีเกิด (birth_year) และจำนวนครั้งที่สามารถยืมสูงสุด (max_loan) โดยสามารถเว้นว่างเพื่อคงค่าของเดิมได้ หลังจากรับข้อมูลแล้ว จะอัปเดตค่าต่าง ๆ ในเรคอร์ดรวมถึงปรับ updated_at เป็นเวลาปัจจุบัน (int(time.time())) เมื่อแก้ไขเสร็จแล้ว เขียนข้อมูลทั้งหมดกลับลงไฟล์ในโหมดเขียนทับ ("wb") และแจ้งผู้ใช้ว่าการแก้ไขสำเร็จ หากไม่พบสมาชิกหรือไม่ใช่ Active จะแจ้งข้อความว่าไม่พบหรือไม่สามารถแก้ไขได้ ทำให้ผู้ใช้สามารถปรับปรุงข้อมูลสมาชิกได้สะดวกและปลอดภัย

```python
def menu_edit_member(filename="members.dat"):
    menu_view_members()
    try:
        member_id = int(input("Enter Member ID to edit: "))
    except ValueError:
        print("\n❌ Invalid input. Please enter a number.")
        return

    members = []
    found = False
    try:
        with open(filename, "rb") as f:
            while True:
                data = f.read(members_struck.size)
                if len(data) < members_struck.size:
                    break
                unpacked = members_struck.unpack(data)
                members.append(list(unpacked))
    except FileNotFoundError:
        print(f"\n❌ File {filename} not found")
        return

    for m in members:
        if m[0] == member_id and m[1] == 1:
            print(f"Editing Member ID {member_id}")
            name = input(f"Enter new Name [{m[2].decode('utf-8').rstrip(chr(0))}]: ")
            try:
                birth_year = input(f"Enter new Birth Year [{m[3]}]: ")
                birth_year = int(birth_year) if birth_year else m[3]
                max_loan = input(f"Enter new Max Loan [{m[4]}]: ")
                max_loan = int(max_loan) if max_loan else m[4]
            except ValueError:
                print("\n❌ Invalid number input. Edit canceled.")
                return

            m[2] = name.encode("utf-8").ljust(50, b"\x00") if name else m[2]
            m[3] = birth_year
            m[4] = max_loan
            m[6] = int(time.time())
            found = True
            break

    if not found:
        print(f"\n❌ Member ID {member_id} not found or not active")
        return

    with open(filename, "wb") as f:
        for m in members:
            record = members_struck.pack(*m)
            f.write(record)

    print(f"\n✅ Member ID {member_id} updated successfully")
```

รูปภาพที่ 4- 18 ฟังก์ชั่น menu_edit_member

4.2.8 ฟังก์ชั่น menu_delete_member

ฟังก์ชัน menu_delete_member ใช้สำหรับลบหรือปิดการใช้งานสมาชิกในไฟล์ members.dat โดยเริ่มจากอ่านข้อมูลสมาชิกทั้งหมดเข้าลิสต์ members ในรูปแบบไบนารีและแปลงเป็น list ของแต่ละเรคอร์ด หากไฟล์ไม่พบจะแจ้งเตือน จากนั้นค้นหาสมาชิกที่ตรงกับรหัส

member_id และมีสถานะ Active (1) หากพบจะเปลี่ยนสถานะเป็น 0 (ปิดการใช้งาน) และปรับค่า updated_at เป็นเวลาปัจจุบัน (int(time.time())) จากนั้นเขียนข้อมูลทั้งหมดกลับลงไฟล์ในโหมดเขียนทับ ("wb") เพื่ออัปเดตข้อมูลในไฟล์ สุดท้ายพิมพ์ข้อความยืนยันว่าการลบสมาชิกสำเร็จ หากไม่พบสมาชิกหรือสมาชิกถูกลบไปแล้ว ฟังก์ชันจะแจ้งผู้ใช้ว่าไม่พบหรือไม่สามารถลบได้ ฟังก์ชันนี้จึงเป็นการลบแบบ soft delete ทำให้ข้อมูลยังอยู่ในไฟล์แต่ระบบจะถือว่าสมาชิกถูกลบเรียบร้อยแล้ว

```python
def menu_delete_member(member_id, filename="members.dat"):
    members = []
    found = False
    try:
        with open(filename, "rb") as f:
            while True:
                data = f.read(members_struck.size)
                if len(data) < members_struck.size:
                    break
                unpacked = members_struck.unpack(data)
                members.append(list(unpacked))
    except FileNotFoundError:
        print(f"\n❌ File {filename} not found")
        return

    for m in members:
        if m[0] == member_id and m[1] == 1:
            m[1] = 0
            m[6] = int(time.time())
            found = True
            break

    if not found:
        print(f"\n❌ Member ID {member_id} not found or already deleted")
        return

    with open(filename, "wb") as f:
        for m in members:
            record = members_struck.pack(*m)
            f.write(record)

    print(f"\n✅ Member ID {member_id} deleted successfully")
```

รูปภาพที่ 4- 19 ฟังก์ชั่น menu_delete_member

4.2.9 ฟังก์ชั่น get_current_loans

ฟังก์ชัน get_current_loans ใช้เพื่อดึงรายการการยืมหนังสือที่ยังไม่คืนจากลิสต์ loans โดยสร้างดิกชันนารี latest เพื่อเก็บรายการยืมล่าสุดของแต่ละคู่ (book_id, member_id) หากมีหลายรายการของคู่เดียวกัน ข้อมูลใหม่จะเขียนทับ ทำให้ได้รายการล่าสุด จากนั้นกรองเฉพาะรายการที่ is_rented_after == 1 ซึ่งหมายถึงหนังสือยังถูกยืมอยู่ และคืนค่าเป็นลิสต์ของรายการการยืมเหล่านี้ ทำให้สามารถตรวจสอบได้ว่ามีสมาชิกใดกำลังยืมหนังสือเล่มใดอยู่

```
def get_current_loans(loans):
    latest = {}
    for loan in loans:
        key = (loan["book_id"], loan["member_id"])
        latest[key] = loan

    current_loans = [l for l in latest.values() if l["is_rented_after"] == 1]
    return current_loans
```

รูปภาพที่ 4- 20 ฟังก์ชั่น get_current_loans

4.2.10 ฟังก์ชั่น menu_borrow_book

ฟังก์ชัน menu_borrow_book ใช้สำหรับให้สมาชิกยืมหนังสือ โดยเริ่มจากดึงข้อมูลหนังสือ สมาชิก และรายการยืมทั้งหมด จากนั้นแสดงรายชื่อหนังสือที่สามารถยืมได้ พร้อมจำนวนเล่มทั้งหมด จำนวนที่ถูกยืม และจำนวนที่ยังว่างอยู่ เพื่อตรวจสอบความพร้อมของหนังสือ ต่อมารับรหัสหนังสือ (book_id) และรหัสสมาชิก (member_id) จากผู้ใช้ และตรวจสอบว่าหนังสือยังมีสถานะ Active และมีสำเนาพอให้ยืม รวมถึงสมาชิกยังมีสถานะ Active หากตรวจสอบผ่าน จะกำหนดวันยืมเป็นปัจจุบัน และวันกำหนดคืนเป็น 30 วันถัดไป แล้วเรียกฟังก์ชัน add_loans เพื่อบันทึกรายการยืมลงไฟล์ loans.dat สุดท้ายจะแจ้งข้อความยืนยันว่าการยืมสำเร็จ พร้อมชื่อสมาชิก ชื่อหนังสือ และวันกำหนดคืน ทำให้กระบวนการยืมหนังสือทั้งตรวจสอบความพร้อมและบันทึกประวัติเป็นไปอย่างครบถ้วนและชัดเจน

```python
def menu_borrow_book():
    print("\n=== Borrow Book ===")
    books = read_all_books("books.dat")
    members = read_all_members("members.dat")

    # แสดงหนังสือที่ยังยืมได้
    print("Available Books:")
    print("-" * 90)
    print(f"{'ID':<6} {'Title':<45} {'Copies':<8} {'Borrowed':<10} {'Available':<10}")
    print("-" * 90)
    loans = read_all_loans("loans.dat")
    current_loans = get_current_loans(loans)
    borrowed_count = {
        b["book_id"]: sum(1 for l in current_loans if l["book_id"] == b["book_id"])
        for b in books
    }
    for b in books:
        if b["status"] == 1:
            borrowed = borrowed_count.get(b["book_id"], 0)
            available = b["copies"] - borrowed
            print(f"{b['book_id']:<6} {b['title']:<45} {b['copies']:<8} {borrowed:<10} {available:<10}")
    print("-" * 90)
    try:
        book_id = int(input("Enter Book ID to borrow: "))
        member_id = int(input("Enter Member ID: "))
    except ValueError:
        print("\n❌ Invalid input. Please enter numbers only.")
        return
    # ตรวจสอบหนังสือ
    book = next((b for b in books if b["book_id"] == book_id and b["status"] == 1), None)
    if not book:
        print(f"\n❌ Book ID {book_id} not found or not active")
        return
    borrowed_now = borrowed_count.get(book_id, 0)
    if borrowed_now >= book["copies"]:
        print("\n❌ No copies available for this book")
        return
    # ตรวจสอบสมาชิก
    member = next((m for m in members if m["member_id"] == member_id and m["status"] == 1), None)
    if not member:
        print(f"\n❌ Member ID {member_id} not found or not active")
        return
    # กำหนดวันยืม/คืน ถัดไป 1 เดือนจากวันที่ยืม
    today = datetime.now().strftime("%Y/%m/%d")
    due_date = (datetime.now() + timedelta(days=30)).strftime("%Y/%m/%d")
    add_loans(
        op_code=1,
        book_id=book_id,
        member_id=member_id,
        loan_date=today,
        return_date=due_date,
        status_after=book["status"],
        is_rented_after=1
    )
    print(f"\n✅ Member '{member['name']}' borrowed '{book['title']}' until {due_date}")
```

รูปภาพที่ 4- 21 ฟังก์ชั่น menu_borrow_book

4.2.11 ฟังก์ชั่น menu_return_book

ฟังก์ชัน menu_return_book ใช้สำหรับบันทึกการคืนหนังสือ โดยเริ่มจากดึงข้อมูลหนังสือ สมาชิก และรายการยืมทั้งหมด จากนั้นกรองรายการยืมที่ยังไม่คืน (current_loans) หากไม่มีรายการยืมจะแจ้งว่า "No books currently borrowed." และออกจากฟังก์ชัน ต่อมาจะแสดงตารางหนังสือที่ยังถูกยืม พร้อมรหัสหนังสือ, ชื่อหนังสือ, รหัสสมาชิก, ชื่อสมาชิก และวันที่ยืม เพื่อให้ผู้ใช้เลือกหนังสือที่ต้องการคืน หลังจากรับรหัสหนังสือและสมาชิกจากผู้ใช้ จะตรวจสอบว่ามีรายการยืมที่ตรงกันหรือไม่ หากไม่พบจะแจ้งข้อความเตือน หากพบ จะกำหนดวันคืนเป็นวันที่ปัจจุบัน และเรียกฟังก์ชัน add_loans เพื่อบันทึกการคืนหนังสือ โดยตั้ง op_code=2 และ is_rented_after=0 เพื่อระบุว่าสิ้นสุดการยืม สุดท้ายจะแจ้งข้อความยืนยันว่าหนังสือถูกคืนเรียบร้อย ทำให้กระบวนการคืนหนังสือครบถ้วนทั้งตรวจสอบความถูกต้องและบันทึกประวัติ

```python
def menu_return_book():
    print("\n=== Return Book ===")
    loans = read_all_loans("loans.dat")
    books = read_all_books("books.dat")
    members = read_all_members("members.dat")
    # แสดงที่ยังไม่คืน
    loans = read_all_loans("loans.dat")
    current_loans = get_current_loans(loans)

    if not current_loans:
        print("No books currently borrowed.")
        return
    print("-" * 97)
    print(f"{'BookID':<8} {'Title':<45} {'MemberID':<10} {'Member Name':<20} {'Loan Date':<10}")
    print("-" * 97)
    for l in current_loans:
        book_title = next((b["title"] for b in books if b["book_id"] == l["book_id"]), "Unknown")
        member_name = next((m["name"] for m in members if m["member_id"] == l["member_id"]), "Unknown")
        print(f"{l['book_id']:<8} {book_title:<45} {l['member_id']:<10} {member_name:<20} {l['loan_date']:<10}")
    print("-" * 97)
    try:
        book_id = int(input("Enter Book ID to return: "))
        member_id = int(input("Enter Member ID: "))
    except ValueError:
        print("\n❌ Invalid input. Please enter numbers only.")
        return
    loan = next((l for l in current_loans if l["book_id"] == book_id and l["member_id"] == member_id), None)
    if not loan:
        print("\n❌ No matching active loan found")
        return
    today = datetime.now().strftime("%Y/%m/%d")
    book = next((b for b in books if b["book_id"] == book_id), None)
    add_loans(
        op_code=2,
        book_id=book_id,
        member_id=member_id,
        loan_date=loan["loan_date"],
        return_date=today,
        status_after=book["status"] if book else 1,
        is_rented_after=0
    )
    print(f"\n✅ Book ID {book_id} has been returned by Member ID {member_id}")
```

รูปภาพที่ 4- 22 ฟังก์ชั่น menu_return_book

4.2.12 ฟังก์ชั่น menu_view_all_loans

ฟังก์ชัน menu_view_all_loans ใช้สำหรับแสดงประวัติการยืม–คืนหนังสือทั้งหมด โดยเริ่มจากดึงข้อมูลรายการยืม หนังสือ และสมาชิกทั้งหมด หากไม่พบรายการยืมจะแจ้งว่า "No loans found." จากนั้นพิมพ์หัวตารางและเส้นคั่นเพื่อจัดรูปแบบให้ชัดเจน สำหรับแต่ละรายการยืม จะดึงชื่อหนังสือและชื่อสมาชิกจากไฟล์ข้อมูลที่เกี่ยวข้อง แปลงรหัสการดำเนินการ (op_code) เป็น "Borrow" หรือ "Return" และสถานะการยืม (is_rented_after) เป็น "Borrowed" หรือ "Returned" แล้วพิมพ์ข้อมูลในรูปแบบตาราง เช่น เวลา, รหัสหนังสือ, ชื่อหนังสือ, รหัสสมาชิก, ชื่อสมาชิก, ประเภทการดำเนินการ และสถานะ ทำให้ผู้ใช้สามารถดูประวัติการยืม–คืนหนังสือทั้งหมดได้อย่างครบถ้วนและชัดเจน

```python
def menu_view_all_loans():
    loans = read_all_loans("loans.dat")
    books = read_all_books("books.dat")
    members = read_all_members("members.dat")

    if not loans:
        print("\nNo loans found.")
        return

    print("-" * 124)
    print(f"{'Timestamp':<20} {'BookID':<7} {'Title':<45} {'MemberID':<10} {'Member Name':<20} {'Type':<8} {'Status':<6}")
    print("-" * 124)

    for loan in loans:
        book_title = next((b["title"] for b in books if b["book_id"] == loan["book_id"]), "Unknown")
        member_name = next((m["name"] for m in members if m["member_id"] == loan["member_id"]), "Unknown")
        loan_type = "Borrow" if loan["op_code"] == 1 else "Return"
        status_text = "Borrowed" if loan["is_rented_after"] == 1 else "Returned"

        print(f"{loan['ts']:<20} {loan['book_id']:<7} {book_title:<45} {loan['member_id']:<10} {member_name:<20} {loan_type:<8} {status_text:<6}")

    print("-" * 124)
```

รูปภาพที่ 4- 23 ฟังก์ชั่น menu_view_all_loans

4.2.12 ฟังก์ชั่น menu_view_current_loans

ฟังก์ชัน menu_view_current_loans ใช้สำหรับแสดงรายการหนังสือที่กำลังถูกยืมอยู่ในปัจจุบัน โดยเริ่มจากอ่านข้อมูลรายการยืมทั้งหมดและกรองเฉพาะรายการที่ยังไม่คืนด้วยฟังก์ชัน get_current_loans หากไม่พบรายการยืมจะแจ้งว่า "No books currently borrowed." จากนั้นดึงข้อมูลหนังสือและสมาชิกทั้งหมดเพื่อให้สามารถแสดงชื่อหนังสือและชื่อสมาชิกได้ครบถ้วน ฟังก์ชันจะแสดงตารางหัวข้อ "Current Loans" พร้อมรหัสหนังสือ, ชื่อหนังสือ, รหัสสมาชิก, ชื่อสมาชิก และวันที่ยืม สำหรับแต่ละรายการยืม ทำให้ผู้ใช้สามารถตรวจสอบได้ว่ามีสมาชิกใดกำลังยืมหนังสือเล่มใดอยู่และเมื่อไหร่ที่ยืม สรุปแล้วฟังก์ชันนี้ช่วยติดตามสถานะการยืมหนังสือได้อย่างชัดเจนและง่ายต่อการตรวจสอบ

```python
def menu_view_current_loans():
    loans = read_all_loans("loans.dat")
    current_loans = get_current_loans(loans)

    if not current_loans:
        print("\nNo books currently borrowed.")
        return

    books = read_all_books("books.dat")
    members = read_all_members("members.dat")

    print("\n=== Current Loans ===")
    print("-" * 97)
    print(f"{'BookID':<8} {'Title':<45} {'MemberID':<10} {'Member Name':<20} {'Loan Date':<10}")
    print("-" * 97)

    for l in current_loans:
        book_title = next((b["title"] for b in books if b["book_id"] == l["book_id"]), "Unknown")
        member_name = next((m["name"] for m in members if m["member_id"] == l["member_id"]), "Unknown")
        print(f"{l['book_id']:<8} {book_title:<45} {l['member_id']:<10} {member_name:<20} {l['loan_date']:<10}")

    print("-" * 97)
```

รูปภาพที่ 4- 24 ฟังก์ชั่น menu_view_current_loans

4.3 เมนู generate_report

4.3.1 ฟังก์ชัน generate_report ใช้สร้างรายงานสรุประบบห้องสมุดลงไฟล์ report.txt โดยเริ่มจากอ่านข้อมูลหนังสือ สมาชิก และรายการยืมทั้งหมด จากนั้นสร้างดิกชันนารี latest_loans เพื่อเก็บรายการยืมล่าสุดของแต่ละคู่ (book_id, member_id) และจัดกลุ่มสมาชิกที่กำลังยืมหนังสืออยู่ (borrowed_now) จากนั้นเปิดไฟล์รายงานในโหมดเขียน (w) และบันทึกข้อมูลหัวรายงาน เช่น เวลาที่สร้างรายงาน เวอร์ชันแอป และรูปแบบการเข้ารหัสดังรูปที่ 4-25

```python
def generate_report(report_file="report.txt"):
    books = read_all_books("books.dat")
    members = read_all_members("members.dat")
    loans = read_all_loans("loans.dat")

    tz = timezone(timedelta(hours=7))
    now = datetime.now(tz).strftime("%Y-%m-%d %H:%M (%z)")

    # --- เตรียม dictionary คนที่ยืม ---
    latest_loans = {}
    for loan in loans:
        key = (loan["book_id"], loan["member_id"])
        latest_loans[key] = loan

    borrowed_now = {}
    for (book_id, member_id), loan in latest_loans.items():
        if loan["is_rented_after"] == 1:
            member_name = next((m["name"] for m in members if m["member_id"] == member_id), "Unknown")
            borrowed_now.setdefault(book_id, []).append(member_name)

    def status_text(status_int):
        return "Active" if status_int == 1 else "Deleted"

    with open(report_file, "w", encoding="utf-8") as f:
        f.write("Library Borrow System - Summary Report\n")
        f.write(f"Generated At : {now} \n")
        f.write("App Version : 2.0\n")
        f.write("Encoding : UTF-8 (fixed-length)\n\n")
```

รูปภาพที่ 4- 25 ฟังก์ชั่น generate_report ใช้ข้อมูลสร้างรายงานสรุป

4.3.2 ต่อมาเขียนตารางหนังสือ โดยแสดงรหัสหนังสือ, ชื่อ, ผู้แต่ง, ปีพิมพ์, จำนวนเล่ม, รายชื่อสมาชิกที่ยืม และสถานะ หากหนังสือถูกยืมหลายคน จะเพิ่มแถวต่อ ๆ ไปเพื่อระบุสมาชิกแต่ละคน จากนั้นสรุปสถิติ เช่น จำนวนหนังสือทั้งหมด, หนังสือ Active/Deleted, หนังสือที่กำลังถูกยืม และจำนวนเล่มว่าง สุดท้ายคำนวณสถิติการยืม เช่น หนังสือที่ถูกยืมมากที่สุด และจำนวนสมาชิก Active ทั้งหมด หลังจากเขียนข้อมูลเสร็จ ฟังก์ชันพิมพ์ข้อความยืนยันว่าได้สร้างรายงานเรียบร้อย ทำให้ผู้ดูแลระบบสามารถตรวจสอบสถานะหนังสือ สมาชิก และสถิติการยืมทั้งหมดได้ในไฟล์เดียวดังรูปที่ 4-26

```
    # -------- ตารางหนังสือ ----------
    for b in books:
        if b["status"] != 1:
            continue

        borrowers = borrowed_now.get(b["book_id"], [])
        if not borrowers:
            f.write(f"{b['book_id']:<6} | {b['title']:<45} | {b['author']:<30} | "
                    f"{b['year']:4d} | {b['copies']:<6} | 0{'':<18} | {status_text(b['status'])}\n")
        else:
            f.write(f"{b['book_id']:<6} | {b['title']:<45} | {b['author']:<30} | "
                    f"{b['year']:4d} | {b['copies']:<6} | 1.{borrowers[0]:<17} | {status_text(b['status'])}\n")
            for i, borrower in enumerate(borrowers[1:], start=2):
                f.write(f"{'':<6} | {'':<45} | {'':<30} | {'':4} | {'':<6} | {i}.{borrower:<17} | \n")

    # -------- ส่วน Summary ----------
    total_books = len(books)
    active_books = sum(1 for b in books if b["status"] == 1)
    deleted_books = total_books - active_books
    borrowed_now_total = sum(len(names) for names in borrowed_now.values())
    available_now = sum(
        (b["copies"] - len(borrowed_now.get(b["book_id"], []))) for b in books if b["status"] == 1
    )

    f.write("\nSummary (Active Books Only)\n")
    f.write(f"- Total Books    : {total_books}\n")
    f.write(f"- Active Books   : {active_books}\n")
    f.write(f"- Deleted Books  : {deleted_books}\n")
    f.write(f"- Borrowed Now   : {borrowed_now_total}\n")
    f.write(f"- Available Now  : {available_now}\n")

    # -------- ส่วน Borrow Statistics ----------
    total_borrowed_times = {b["book_id"]: 0 for b in books}
    for loan in loans:
        if loan["op_code"] == 1:
            total_borrowed_times[loan["book_id"]] += 1

    if total_borrowed_times:
        most_borrowed_book_id = max(total_borrowed_times, key=total_borrowed_times.get, default=None)
        if most_borrowed_book_id is not None:
            most_borrowed_times = total_borrowed_times[most_borrowed_book_id]
            most_borrowed_title = next((b["title"] for b in books if b["book_id"] == most_borrowed_book_id), "N/A")
        else:
            most_borrowed_title = "N/A"
            most_borrowed_times = 0
    else:
        most_borrowed_title = "N/A"
        most_borrowed_times = 0

    active_members = sum(1 for m in members if m["status"] == 1)

    f.write("\nBorrow Statistics (Active only)\n")
    f.write(f"- Most Borrowed Book : {most_borrowed_title} ({most_borrowed_times} times)\n")
    f.write(f"- Currently Borrowed : {borrowed_now_total}\n")
    f.write(f"- Active Members     : {active_members}\n")

print(f"\n✅ Report generated: {report_file}")
```

รูปภาพที่ 4- 26 ฟังก์ชั่น generate_report ใช้สร้างไฟล์

4.4 main_menu ระบบยืม - คืนหนังสือห้องสมุด

4.4.1 ฟังก์ชัน main_menu เป็นเมนูหลักของระบบยืม–คืนหนังสือ โดยทำงานในลูปไม่สิ้นสุด (while True) เพื่อให้ผู้ใช้สามารถเลือกทำงานหลายครั้ง เริ่มจากแสดงหัวข้อ "Library Borrow System" พร้อมตัวเลือก ได้แก่ 1. จัดการหนังสือ 2. จัดการสมาชิก 3. จัดการการยืม–คืน 4. สร้างรายงาน และ 5. ออกจากโปรแกรม หลังจากผู้ใช้ป้อนตัวเลือก ฟังก์ชันจะเรียกฟังก์ชันย่อยที่เกี่ยวข้องตามตัวเลือก เช่น manage_books(), manage_members(), manage_loans() หรือ generate_report("report.txt") หากผู้ใช้เลือกออก (5) จะสร้างรายงานก่อนออกและแสดงข้อความ "Exiting program..." หากผู้ใช้ป้อนค่าที่ไม่ถูกต้อง จะพิมพ์ข้อความเตือน "Invalid option! Please select 1-5." ทำให้เมนูหลักนี้ช่วยให้ผู้ใช้เข้าถึงฟังก์ชันต่าง ๆ ของระบบได้สะดวกและควบคุมโปรแกรมได้ง่าย

```python
def main_menu():
    while True:
        print("\n=== Library Borrow System ===")
        print("1. Manage Books")
        print("2. Manage Members")
        print("3. Manage Loans")
        print("4. Generate report")
        print("5. Exit")
        choice = input("Select an option (1-5): ")

        if choice == "1":
            manage_books()
        elif choice == "2":
            manage_members()
        elif choice == "3":
            manage_loans()
        elif choice == "4":
            generate_report("report.txt")
        elif choice == "5":
            generate_report("report.txt")
            print("\nExiting program...")
            break
        else:
            print("\n❌ Invalid option! Please select 1-5.")
```

รูปภาพที่ 4- 27 main_menu

4.5 เมนู Book

4.5.1 ฟังก์ชัน manage_books เป็นเมนูย่อยสำหรับจัดการข้อมูลหนังสือ ทำงานในลูปไม่สิ้นสุดเพื่อให้ผู้ใช้ทำงานหลายครั้ง เริ่มจากแสดงหัวข้อ "Manage Books" พร้อมตัวเลือก ได้แก่ 1. เพิ่มหนังสือ 2. แสดงรายการหนังสือทั้งหมด 3. แก้ไขหนังสือ 4. ลบหนังสือ และ 5. กลับไปเมนูหลัก หลังจากผู้ใช้ป้อนตัวเลือก ฟังก์ชันจะเรียกเมนูย่อยที่เกี่ยวข้อง เช่น menu_add_book(), menu_view_books(), menu_edit_book() สำหรับลบหนังสือ จะเรียก menu_view_books() เพื่อแสดงรายการทั้งหมดก่อน จากนั้นรับรหัสหนังสือ (book_id) และเรียก menu_delete_book (book_id) โดยตรวจสอบความถูกต้องของค่าที่ป้อน หากเลือก 5 จะออกจากลูปและกลับไปเมนูหลัก หากป้อนตัวเลือกไม่ถูกต้องจะแจ้งเตือนผู้ใช้ ทำให้การจัดการหนังสือสะดวกและควบคุมได้ง่าย

```python
def manage_books():
    while True:
        print("\n--- Manage Books ---")
        print("1. Add Book")
        print("2. View All Books")
        print("3. Edit Book")
        print("4. Delete Book")
        print("5. Back to Main Menu")
        choice = input("Select an option (1-5): ")

        if choice == "1":
            menu_add_book()
        elif choice == "2":
            menu_view_books()
        elif choice == "3":
            menu_edit_book()
        elif choice == "4":
            menu_view_books()
            while True:
                try:
                    book_id = int(input("Enter Book ID: "))
                    menu_delete_book(book_id)
                    break
                except ValueError:
                    print("\n❌ Invalid input. Please enter a number.")
        elif choice == "5":
            break
        else:
            print("\n❌ Invalid option! Please select 1-5.")
```

รูปภาพที่ 4- 28 เมนู Book

4.6 เมนู Members

4.6.1 ฟังก์ชัน manage_members เป็นเมนูย่อยสำหรับจัดการข้อมูลสมาชิก ทำงานในลูปไม่สิ้นสุดเพื่อให้ผู้ใช้สามารถทำงานหลายครั้ง โดยแสดงตัวเลือกให้ผู้ใช้ ได้แก่ เพิ่มสมาชิก, แสดงรายการสมาชิกทั้งหมด, แก้ไขสมาชิก, ลบสมาชิก และกลับไปเมนูหลัก หลังจากผู้ใช้ป้อนตัวเลือก ฟังก์ชันจะเรียกเมนูย่อยที่เกี่ยวข้อง เช่น menu_add_member(), menu_view_members(), menu_edit_member() สำหรับการลบสมาชิก จะเรียก menu_view_members() เพื่อแสดงรายการสมาชิกทั้งหมดก่อนรับสมัครสมาชิก (member_id) และเรียก menudelete_member (member_id) พร้อมตรวจสอบความถูกต้องของข้อมูล หากผู้ใช้เลือกกลับไปเมนูหลัก (5) ฟังก์ชันจะออกจากลูป และหากป้อนตัวเลือกไม่ถูกต้องจะแจ้งเตือน ทำให้การจัดการสมาชิกสะดวกและควบคุมได้ง่าย

```python
def manage_members():
    while True:
        print("\n--- Manage Members ---")
        print("1. Add Member")
        print("2. View All Members")
        print("3. Edit Member")
        print("4. Delete Member")
        print("5. Back to Main Menu")

        choice = input("Select an option (1-5): ")
        if choice == "1":
            menu_add_member()
        elif choice == "2":
            menu_view_members()
        elif choice == "3":
            menu_edit_member()
        elif choice == "4":
            menu_view_members()
            while True:
                try:
                    member_id = int(input("Enter Member ID: "))
                    menu_delete_member(member_id)
                    break
                except ValueError:
                    print("\n❌ Invalid input. Please enter a number.")
        elif choice == "5":
            break
        else:
            print("\n❌ Invalid option! Please select 1-5.")
```

รูปภาพที่ 4- 29 เมนู Members

4.7 เมนู Loans

4.7.1 ฟังก์ชัน manage_loans เป็นเมนูย่อยสำหรับจัดการการยืม – คืนหนังสือ ทำงานในลูปไม่สิ้นสุดเพื่อให้ผู้ใช้สามารถทำงานหลายครั้ง โดยแสดงตัวเลือกให้ผู้ใช้ ได้แก่ ยืมหนังสือ, คืนหนังสือ, แสดงรายการการยืมทั้งหมด, แสดงรายการการยืมปัจจุบัน และกลับไปเมนูหลัก หลังจากผู้ใช้ป้อนตัวเลือก ฟังก์ชันจะเรียกเมนูย่อยที่เกี่ยวข้อง เช่น menu_borrow_book(), menu_return_book(), menu_view_all_loans(), menu_view_current_loans() หากเลือกกลับไปเมนูหลัก (5) ฟังก์ชันจะออกจากลูป และหากป้อนตัวเลือกไม่ถูกต้องจะแจ้งเตือนผู้ใช้ ทำให้การจัดการการยืม–คืนหนังสือสะดวกและควบคุมได้ง่าย

```python
def manage_loans():
    while True:
        print("\n--- Manage Loans ---")
        print("1. Borrow Book")
        print("2. Return Book")
        print("3. View All Loans")
        print("4. Current Loans")
        print("5. Back to Main Menu")

        choice = input("Select an option (1-5): ")
        if choice == "1":
            menu_borrow_book()
        elif choice == "2":
            menu_return_book()
        elif choice == "3":
            menu_view_all_loans()
        elif choice == "4":
            menu_view_current_loans()
        elif choice == "5":
            break
        else:
            print("\n❌ Invalid option! Please select 1-5.")
```

รูปภาพที่ 4- 30 เมนู Loans

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

ระบบยืม – คืนหนังสือห้องสมุด ที่พัฒนาขึ้นสามารถช่วยจัดการข้อมูลหนังสือ ข้อมูลสมาชิก และข้อมูลการยืม – คืนได้อย่างมีประสิทธิภาพ โดยใช้การจัดเก็บข้อมูลแบบไฟล์ไบนารี พร้อมเมนูสำหรับเพิ่ม แก้ไข ลบ และแสดงผลข้อมูล ระบบยังรองรับการตรวจสอบสถานะหนังสือที่ถูกยืมหรือยังว่างอยู่ การควบคุมจำนวนเล่มที่ถูกยืม ตลอดจนการสร้างรายงานสรุปผลการดำเนินงาน เช่น จำนวนหนังสือที่ถูกยืมมากที่สุด รายชื่อผู้ยืมปัจจุบัน และสถิติการใช้งานโดยรวม ซึ่งช่วยให้การบริหารจัดการห้องสมุดสะดวก รวดเร็ว และลดความผิดพลาดจากการบันทึกแบบเดิมที่ใช้เอกสารกระดาษ

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

ในการพัฒนาระบบยืม – คืนหนังสือห้องสมุด พบปัญหาหลักคือ ความซับซ้อนของการจัดการไฟล์ไบนารีที่ต้องใช้โครงสร้างข้อมูลคงที่ (struct) ซึ่งอาจเกิดข้อผิดพลาดหากการเข้ารหัสหรือถอดรหัสไม่ถูกต้อง นอกจากนี้ยังพบข้อจำกัดด้านการแสดงผลข้อมูล เช่น ความยาวของชื่อหนังสือหรือชื่อผู้ใช้ที่ต้องถูกจำกัดตามขนาดที่กำหนดไว้ อีกทั้งระบบยังไม่มีการเชื่อมต่อฐานข้อมูลจริง ทำให้การจัดการข้อมูลจำนวนมากหรือการเข้าถึงพร้อมกันจากหลายผู้ใช้งานยังไม่สามารถทำได้เต็มที่

5.3 ข้อเสนอแนะ

เพื่อให้ระบบสมบูรณ์และพร้อมใช้งานจริงในอนาคต ควรปรับปรุงดังนี้

5.3.1 พัฒนาให้รองรับฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เช่น MySQL หรือ SQLite เพื่อรองรับข้อมูลจำนวนมากและการเข้าถึงหลายผู้ใช้งาน

5.3.2 เพิ่มฟังก์ชันค้นหาและกรองข้อมูล เช่น ค้นหาหนังสือตามชื่อ ผู้แต่ง หรือปีที่พิมพ์

5.3.3 ปรับปรุงระบบยืนยันตัวตนสมาชิก และจำกัดสิทธิ์การเข้าถึงของผู้ใช้แต่ละกลุ่ม

5.3.4 พัฒนาเป็นโปรแกรมที่มีส่วนติดต่อผู้ใช้แบบกราฟิก (GUI) หรือเว็บแอปพลิเคชัน เพื่อความสะดวกในการใช้งานจริง

5.4 สิ่งที่ผู้จัดทำได้รับในการพัฒนาโครงงาน

จากการพัฒนาโครงงานครั้งนี้ ผู้จัดทำได้รับความรู้และประสบการณ์ด้านการออกแบบระบบ การเขียนโปรแกรมด้วยภาษา Python การใช้โครงสร้างข้อมูลแบบไบนารี รวมถึงการคิดวิเคราะห์และแก้ไขปัญหาเชิงตรรกะ นอกจากนี้ยังได้ฝึกทักษะการทำงานเป็นทีม การแบ่งหน้าที่รับผิดชอบ และการจัดการเวลาให้สอดคล้องกับแผนงาน ทำให้ผู้จัดทำมีความเข้าใจในกระบวนการพัฒนาระบบซอฟต์แวร์มากยิ่งขึ้น และสามารถนำไปประยุกต์ใช้ในโครงการหรืองานจริงในอนาคตได้