รายงาน

Building a Library Management System

โดย

นางสาวปาณิศา กาญจนเพิ่มพูน รหัสนักศึกษา 630910176

นายโชคทวี ศรศิลป์ รหัสนักศึกษา 630910315

นายโชคทวี ฟ้าคนอง รหัสนักศึกษา 630910316

นายตะวัน ไชยมาตร์ รหัสนักศึกษา 630910323

เสนอ

อาจารย์ศักดิ์ระพี ไพศาลนันทน์

สาขาวิศวกรรมอิเล็กทรอนิกส์และระบบคอมพิวเตอร์

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยศิลปากร

สารบัญ

บทที่ 1 บทนำ

1.วัตถุประสงค์

เพื่อให้นักศึกษาได้ศึกษาค้นคว้าวิธีการพัฒนาแอพลิเคชั่นโดยชั่นหลักการเขียนโปรแกรมเชิงวัตถุ (OPP) เพื่อจัดการระบบห้องสมุด หรือแอพลิเคชั่นอื่นๆได้

2.โปรแกรมที่ใช้งาน

- Python
- Visual Studio Code
- PySimpleGUI

3.เทคนิคที่ใช้เขียนโปรแกรม

1) Encapsulation
2) Polymorphism
3) Inheritance

4.ลักษณะงานที่ได้

1) Classes

= Member, Library, Publication, Book และ Loan

2) Functionality

Member Management

- เก็บข้อมูลของสมาชิกในห้องสมุด (ชื่อ, รหัสสมาชิก, เบอร์)
- เก็บข้อมูลพื้นฐานของหนังสือ (ชื่อ, ผู้เขียน, ปีที่ตีพิมพ์)
- เก็บข้อมูลชนิดหนังสือ
- เก็บข้อมูลการยืมและการจัดการยืมต่างๆ (อัพเดตต่างๆ)
- เพิ่มสมาชิกใหม่ และ แสดงลายละเอียด

Publication Management

- โดเนทหนังสือ

- ค้นหาหนังสือที่ได้รับมา (โดเนท)

- อัพเดตรายการหนังสือ

Loan Management

- ยืมหนังสือโดยสมาชิก

- แสดงรายละเอียดการยืม

- Report

- ส่งช้ากว่ากำหนด

3) User Interface

- ใช้ PySimpleGUI ในการสร้างอินเตอร์เฟส

บทที่ 2 ขั้นตอนการทำงาน

ในการเขียนโปรแกรมจัดการระบบห้องสมุดจะแบ่งเป็น 2 ส่วน

1.ส่วนของ Main

1) ติดตั้ง PySimpleGUI ด้วย pip install PySimpleGUI

2) สร้างและตั้งชื่อไฟล์ LIB.py

3) เพิ่มโมดูล datetime และ timedelta เพื่อจัดการส่วนของวันเวลา

4) สร้างคลาสและฟังก์ชันออกมาได้แก่ Member, Library, Publication, Book และ Loan

- Member เก็บข้อมูลของสมาชิกในห้องสมุด เช่น ชื่อ (name), รหัสสมาชิก (member_id), และข้อมูลติดต่อ (contact_info)

- Library เก็บข้อมูลหนังสือและสมาชิก รวมถึงการจัดการการยืมหนังสือด้วยเมธอดต่าง ๆ เช่น add_member() เพื่อเพิ่มสมาชิก, add_publication()

- Publication เก็บข้อมูลพื้นฐานของหนังสือ ชื่อ (title), ผู้เขียน (author), และปีที่ตีพิมพ์ (year). มีเมธอด additional_info()

- Book มีคุณสมบัติเพิ่มเติมเกี่ยวกับหนังสือ เช่น ISBN (isbn) และประเภทของหนังสือ (book_type) นอกจากนี้ยังมีเมธอด additional_info()

- Loan ใช้เก็บข้อมูลเกี่ยวกับการยืมหนังสือ มีคุณสมบัติดังนี้: สมาชิก (member), การตีพิมพ์ (publication), วันที่ยืม (borrow_date), และวันที่กำหนดคืน (due_date).

2. ส่วนของ GUI

1) สร้างและตั้งชื่อไฟล์ GUI.py

2) เพิ่มไลบารี่ PySimpleGUI,คลาส LIB, โมดูล datetime และ timedelta

3) สร้างฟังก์ชันต่างๆดังนี้

- ฟังก์ชัน add_new_member_gui(library) สร้าง GUI สำหรับเพิ่มสมาชิกใหม่เข้าสู่ระบบของห้องสมุด

- ฟังก์ชัน search_member_gui(library) สร้าง GUI สำหรับค้นหาสมาชิกโดยให้ผู้ใช้ป้อนรหัสสมาชิกและแสดงข้อมูลสมาชิก

- ฟังก์ชัน add_new_book_gui(library) สร้าง GUI สำหรับเพิ่มหนังสือใหม่เข้าสู่ระบบของห้องสมุด โดยให้ผู้ใช้ป้อนข้อมูลเช่น ชื่อหนังสือ, ผู้แต่ง, ปีที่ตีพิมพ์, ISBN, และประเภทหนังสือ

- ฟังก์ชัน ฟังก์ชัน search_book_gui(library) สร้าง GUI สำหรับค้นหาหนังสือโดยให้ผู้ใช้ป้อนชื่อหนังสือและแสดงข้อมูลของหนังสือที่พบหรือแจ้งว่าไม่พบหนังสือนั้น

- ฟังก์ชัน borrow_book_gui(library) สร้าง GUI สำหรับยืมหนังสือโดยให้ผู้ใช้ป้อนรหัสสมาชิกและชื่อหนังสือที่ต้องการยืม เมื่อผู้ใช้กดปุ่ม "Borrow Book" ระบบจะทำการยืมหนังสือและบันทึกข้อมูลการยืม

- ฟังก์ชัน display_loan_details_gui(library) สร้าง GUI สำหรับแสดงรายละเอียดการยืมของสมาชิกโดยให้ผู้ใช้ป้อนรหัสสมาชิก และแสดงรายการหนังสือที่สมาชิกนั้นยืมไว้

- ฟังก์ชัน display_overdue_books_gui(library) สร้าง GUI สำหรับแสดงหนังสือที่เลยกำหนดคืนแล้ว

บทที่ 3 ขั้นตอนการปฏิบัติงาน

ส่วนของโค้ดโปรแกรม

1. ส่วนของ LIB.py

```
from datetime import datetime, timedelta

class Member:
    def __init__(self, name, member_id, contact_info): #สร้างอ็อบเจกต์ของคลาส Member โดยกำหนดค่าเริ่มต้นกับข้อมูลพื้นฐานเกี่ยวกับสมาชิกของห้องสมุด

        self.name = name
        self.member_id = member_id
        self.contact_info = contact_info

class Publication:
    def __init__(self, title, author, year):
        self.title = title
        self.author = author
        self.year = year

    def additional_info(self):#เมธอดใช้สำหรับการเพิ่มข้อมูลเพิ่มเติมเกี่ยวกับหนังสือ
        pass

class Book(Publication):
    def __init__(self, title, author, year, isbn, book_type): #สร้างอ็อบเจกต์ของคลาส Book โดยกำหนดค่าเริ่มต้นของแต่ละแอตทริบิวต์
```

```
        super().__init__(title, author, year)
        self.isbn = isbn
        self.book_type = book_type

    def additional_info(self):#คืนข้อมูลเพิ่มเติมของหนังสือ
        return f"ISBN: {self.isbn}, Type: {self.book_type}"

class Loan:
    def __init__(self, member, publication, borrow_date, due_date): #สร้างอ็อบเจกต์ของคลาส Loan เพื่อเก็บข้อมูลการยืมหนังสือ

        self.member = member
        self.publication = publication
        self.borrow_date = borrow_date
        self.due_date = due_date

class Library:
    def __init__(self):
        self.members = [] #เก็บข้อมูลของสมาชิกทั้งหมดในห้องสมุด
        self.publications = [] #เก็บข้อมูลของหนังสือทั้งหมดในห้องสมุด
        self.loans = [] #เก็บข้อมูลการยืมหนังสือที่ทำไว้ในห้องสมุด

    def add_member(self, member): #เมธอดนี้ใช้สำหรับเพิ่มสมาชิกเข้าไปในรายชื่อสมาชิกของกลุ่ม
        self.members.append(member)

    def find_member(self, member_id): #ใช้ลูป for เพื่อวนลูปผ่านสมาชิกทุกตัวในลิสต์ self.members
        for member in self.members:  #ตรวจสอบว่า member_id ของสมาชิกในแต่ละรอบของลูปเท่ากับ member_id ที่เราต้องการค้นหาหรือไม่
            if member.member_id == member_id: #หากพบสมาชิกที่ตรงกับ member_id ที่เราต้องการค้นหา โค้ดจะคืนค่าสมาชิกนั้นออกมา
                return member  #หากไม่พบสมาชิกที่ตรงกับ member_id ที่เราต้องการค้นหา โค้ดจะคืนค่า None
        return None

    def display_member_details(self, member): #แสดงรายละเอียดของสมาชิกที่รับเข้ามา
        print(f"Name: {member.name}")
        print(f"Student ID: {member.member_id}")
        print(f"Phone Number: {member.contact_info}")

    def add_publication(self, publication): #เพิ่มข้อมูลการตีพิมพ์เข้าไปในรายการ
        self.publications.append(publication)

    def find_publication(self, title): #วนลูปผ่านรายการการตีพิมพ์ทั้งหมด
        for pub in self.publications: #ตรวจสอบว่าชื่อหนังสือของแต่ละรายการตรงกับชื่อที่ระบุหรือไม่
            if pub.title == title: #หากพบรายการการตีพิมพ์ที่ตรงกับชื่อที่ระบุ ให้คืนค่ารายการตีพิมพ์นั้น
                return pub
```

```
        return None  #ถ้าไม่พบรายการการตีพิมพ์ที่ตรงกับชื่อที่ระบุ ให้คืนค่า None

    def display_publication_details(self, publication): #แสดงรายละเอียดของการตีพิมพ์ที่ระบุ
        print(f"Title: {publication.title}")
        print(f"Author: {publication.author}")
        print(f"Year: {publication.year}")
        print(publication.additional_info())

    def borrow_publication(self, member, publication, borrow_date, due_date): #
ตรวจสอบว่าการตีพิมพ์ที่ต้องการยืมอยู่ในรายการหรือไม่
        if publication in self.publications: #หากใช่ เพิ่มข้อมูลการยืมลงในรายการยืม
            self.loans.append(Loan(member, publication, borrow_date, due_date))
            return True
        else: #หากไม่ใช่ คืนค่า False เพื่อแสดงว่าไม่สามารถยืมได้
            return False

    def is_book_overdue(self, book): #วนลูปผ่านรายการยืมทั้งหมด
        for loan in self.loans: #ตรวจสอบว่าการตีพิมพ์ที่ถูกยืมในรายการยืมเป็นหนังสือที่เราต้องการตรวจสอบหรือไม่
            if loan.publication == book: #แปลงวันที่กำหนดส่งคืนของการยืมเป็นวัตถุ datetime
                due_date = datetime.strptime(loan.due_date, '%Y-%m-%d') # เปรียบเทียบว่า
วันปัจจุบันมากกว่าวันที่กำหนดส่งคืนหรือไม่
                if datetime.now() > due_date:
                    return True
        return False

    def display_overdue_books(self): #สร้างรายการเพื่อเก็บหนังสือที่เกินกำหนด
        overdue_books = [] #วนลูปผ่านรายการยืมทั้งหมด
        for loan in self.loans: #แปลงวันที่กำหนดส่งคืนให้เป็นวัตถุ datetime
            due_date = datetime.strptime(loan.due_date, '%Y-%m-%d')  #ตรวจสอบว่าหนังสือ
เกินกำหนดหรือไม่
            if datetime.now() > due_date: #หากเกินกำหนด เพิ่มหนังสือลงในรายการหนังสือที่เกินกำหนด
                overdue_books.append(loan.publication)

        if overdue_books: #วนลูปผ่านหนังสือที่เกินกำหนด
            print("Overdue Books:") #ถ้ามีหนังสือที่เกินกำหนดโค้ดจะแสดงข้อความ หนังสือที่เกินกำหนด
            for book in overdue_books:
                print(f"- {book.title} by {book.author}")
        else:
            print("No overdue books found.")  #ถ้าไม่มีหนังสือที่เกินกำหนดโค้ดจะแสดงข้อความ ไม่พบหนังสือที่เกิน
กำหนด

    def display_loan_details(self, member):  #ค้นหารายการยืมทั้งหมดของสมาชิกที่กำหนด
        loans = [loan for loan in self.loans if loan.member == member]
        if loans:
            print("Trask Books:")
```

```
            for loan in loans: #วนลูปผ่านรายการยืมและแสดงรายละเอียด
                print(f"Publication: {loan.publication.title}")
                print(f"Borrow Date: {loan.borrow_date}")
                print(f"Due Date: {loan.due_date}")
        else:
            print("No Tasks Found For This Student.")

def add_new_member(library): #รับข้อมูลนักเรียนใหม่เข้าไปในห้องสมุด
    name = input("Enter Student's name: ")
    member_id = input("Enter Student ID: ")
    contact_info = input("Enter Phone Number: ")
    new_member = Member(name, member_id, contact_info)
    library.add_member(new_member)  #เพิ่มนักเรียนใหม่เข้าไปในห้องสมุด
    print("New Student Added Successfully.") #ข้อมูลนักเรียนใหม่เข้าไปในห้องสมุดเรียบร้อบแล้ว

def search_member(library): #รับรหัสนักเรียนที่ต้องการค้นหา
    member_id = input("Enter Student ID: ")  #ค้นหาข้อมูลของนักเรียนจากรหัสนักเรียนที่ระบุ
    member = library.find_member(member_id)
    if member:
        library.display_member_details(member) # แสดงรายละเอียดของนักเรียน
    else:
        print("Student not found.")

def add_new_book(library): #รับข้อมูลหนังสือใหม่
    title = input("Enter book title: ")
    author = input("Enter author: ")
    year = input("Enter year of publication: ")
    isbn = input("Enter ISBN: ")
    book_type = input("Enter book type: ")
    new_book = Book(title, author, year, isbn, book_type) #เพิ่มหนังสือใหม่เข้าไปในห้องสมุด
    library.add_publication(new_book)
    print("New book added successfully.") #เพิ่มหนังสือใหม่เข้าห้องสมุดเรียบร้อยแล้ว

def search_book(library): #รับชื่อหนังสือที่ต้องการค้นหา
    title = input("Enter book title: ") #ค้นหาข้อมูลของหนังสือจากชื่อที่ระบุ
    book = library.find_publication(title)
    if book:
        library.display_publication_details(book)  #แสดงรายละเอียดของหนังสือ
    else:
        print("Book not found.")

def borrow_book(library): #รับรหัสสมาชิกที่ต้องการยืมหนังสือ
    member_id = input("Enter member ID: ") #ค้นหาข้อมูลของสมาชิกจากรหัสที่ระบุ
    member = library.find_member(member_id)
    if not member: #ถ้าไม่พบสมาชิก
```

```
        print("Member not found.") #แสดงข้อความว่า "ไม่พบสมาชิก" และสิ้นสุดการทำงานของเมธอด
        return

    title = input("Enter book title to borrow: ") #รับชื่อหนังสือที่ต้องการยืมจากผู้ใช้
    book = library.find_publication(title) #เมธอด find_publication ของห้องสมุดเพื่อค้นหาข้อมูลของหนังสือจากชื่อที่ระบุ
    if not book:
        print("Book not found.")
        return

    borrow_date = datetime.now().strftime('%Y-%m-%d') #ยืมหนังสือโดยกำหนดวันที่ยืมและกำหนดวันคืน
    due_date = (datetime.now() + timedelta(days=14)).strftime('%Y-%m-%d')  # เพิ่มระยะเวลายืม 14 วัน
    if library.borrow_publication(member, book, borrow_date, due_date):
        print("Book borrowed successfully.")
    else:
        print("Book is not available for borrowing.")

def display_loan_details(library): #แสดงรายละเอียดของการยืมหนังสือโดยรับรหัสสมาชิกจากผู้ใช้
    member_id = input("Enter member ID: ") #ค้นหาข้อมูลของสมาชิกจากรหัสที่ระบุ
    member = library.find_member(member_id)
    if member:
        library.display_loan_details(member)  #แสดงรายละเอียดของการยืมของสมาชิก
    else:
        print("Member not found.")

def main(): #ฟังก์ชัน main() ดังกล่าวเป็นจุดเริ่มต้นของโปรแกรม โดยมีลักษณะเป็นการแสดงเมนูให้ผู้ใช้เลือกทำรายการต่าง ๆ
    library = Library()
    while True:
        print("\n♦♦♦♦♦♦♦ WELCOME TO THE Mahalnw LIBRARY ♦♦♦♦♦♦♦\n")
        print("1. Add New Student")
        print("2. Search Student Data")
        print("3. Donate Book")
        print("4. Search Book")
        print("5. Borrow Book")
        print("6. Tasks Books")
        print("7. Overdue Books")
        print("8. Exit")

        choice = input("Enter your choice: ")
        #ใช้สำหรับการดำเนินการตามที่ผู้ใช้ได้ทำเลือกในเมนู โดยเช็คค่า choice ที่ผู้ใช้ป้อนเข้ามา และดำเนินการตามเงื่อนไขต่าง ๆ
        if choice == '1':
            add_new_member(library)
        elif choice == '2':
            search_member(library)
```

```
        elif choice == '3':
            add_new_book(library)
        elif choice == '4':
            search_book(library)
        elif choice == '5':
            borrow_book(library)
        elif choice == '6':
            display_loan_details(library)
        elif choice == '7':
            library.display_overdue_books()
        elif choice == '8':
            print("Exiting Program.")
            break
        else:
            print("Invalid Choice. Please Try Again.")


if __name__ == "__main__":
    main()
```

2. ส่วนของ GUI.py

```
import PySimpleGUI as sg # เพิ่มไลบรี่ PySimpleGUI
from datetime import datetime, timedelta # เพิ่มโมดูล datetime, timedelta
from LIB import Library, Member, Book # เพิ่มคลาส LIB.py

# ฟังก์ชันสำหรับเพิ่มสมาชิกใหม่
def add_new_member_gui(library): #กำหนดโครงร่างของหน้าต่าง GUI
    layout = [
        [sg.Text('Enter Student\'s name: '), sg.InputText(key='-NAME-')],
        [sg.Text('Enter Student ID: '), sg.InputText(key='-ID-')],
        [sg.Text('Enter Phone Number: '), sg.InputText(key='-CONTACT-')],
        [sg.Button('Add Student'), sg.Button('Cancel')]
    ]
    window = sg.Window('Add New Student', layout) #สร้างหน้าต่าง GUI สำหรับเพิ่มนักเรียนใหม่
    while True:
        event, values = window.read() #รอรับเหตุการณ์และข้อมูลที่กรอกเข้ามาในหน้าต่าง GUI
        if event == sg.WINDOW_CLOSED or event == 'Cancel':
            break #ออกจาก loop และจบการทำงานของโปรแกรม
        elif event == 'Add Student':
            name = values['-NAME-']
            member_id = values['-ID-']
            contact_info = values['-CONTACT-'] #สร้างอ็อบเจ็กต์ Member จากข้อมูลที่กรอก
            new_member = Member(name, member_id, contact_info) #สร้างอ็อบเจ็กต์ Member จาก
ข้อมูลที่กรอก
```

```
            library.add_member(new_member)
            sg.popup("New Student Added Successfully.")  #แสดงข้อความยืนยันการเพิ่มนักเรียนใหม่
            break #ออกจาก loop และจบการทำงานของโปรแกรม
    window.close() #ปิดหน้าต่าง GUI หลังจากเสร็จสิ้นการทำงาน


# ฟังก์ชันสำหรับค้นหาสมาชิก
def search_member_gui(library):  #กำหนดโครงร่างของหน้าต่าง GUI
    layout = [
        [sg.Text('Enter Student ID: '), sg.InputText(key='-ID-')],
        [sg.Button('Search'), sg.Button('Cancel')]
    ]
    window = sg.Window('Search Student Data', layout) #สร้างหน้าต่าง GUI สำหรับค้นหาข้อมูลนักเรียน
    while True:
        event, values = window.read() #รอรับเหตุการณ์และข้อมูลที่กรอกเข้ามาในหน้าต่าง GUI
        if event == sg.WINDOW_CLOSED or event == 'Cancel':
            break #ออกจาก loop และจบการทำงานของโปรแกรม
        elif event == 'Search':
            member_id = values['-ID-']
            member = library.find_member(member_id)  #เรียกใช้เมธอด find_member ในคลาส
Library เพื่อค้นหาข้อมูลนักเรียน
            if member:
                sg.popup(f"Name: {member.name}\nMember ID:
{member.member_id}\nContact Info: {member.contact_info}") #เมื่อพบข้อมูลของนักเรียนโปรแกรมจะแสดง
ข้อมูลของนักเรียนนั้นในหน้าต่าง Popup
            else:
                sg.popup("Member not found.")
            break #ออกจาก loop และจบการทำงานของโปรแกรม
    window.close() #โปรแกรมออกจากลูปและสิ้นสุดการทำงาน และปิดหน้าต่าง GUI ด้วยคำสั่ง window.close()

# ฟังก์ชันสำหรับเพิ่มหนังสือใหม่
def add_new_book_gui(library): #สร้างหน้าต่าง GUI โดยใช้โครงร่าง layout ที่กำหนด
    layout = [
        [sg.Text('Enter Book Title: '), sg.InputText(key='-TITLE-')],
        [sg.Text('Enter Author: '), sg.InputText(key='-AUTHOR-')],
        [sg.Text('Enter Year of Publication: '), sg.InputText(key='-YEAR-')],
        [sg.Text('Enter ISBN: '), sg.InputText(key='-ISBN-')],
        [sg.Text('Enter Book Type: '), sg.InputText(key='-TYPE-')],
        [sg.Button('Donate Book'), sg.Button('Cancel')]
    ]
    window = sg.Window('Donate Book', layout) #สร้างหน้าต่าง GUI สำหรับบริจาคหนังสือใหม่
    while True:
        event, values = window.read() #รอรับเหตุการณ์และข้อมูลที่กรอกเข้ามาในหน้าต่าง GUI
        if event == sg.WINDOW_CLOSED or event == 'Cancel':
            break #ออกจาก loop และจบการทำงานของโปรแกรม
```

```
        elif event == 'Donate Book':
            title = values['-TITLE-'] #ดึงข้อมูลชื่อหนังสือที่ผู้ใช้กรอกเข้ามา
            author = values['-AUTHOR-']
            year = values['-YEAR-']
            isbn = values['-ISBN-']
            book_type = values['-TYPE-'] #ดึงข้อมูลประเภทหนังสือ
            new_book = Book(title, author, year, isbn, book_type) #สร้างอ็อบเจกต์ของหนังสือ
ใหม่ด้วยข้อมูลที่ได้รับ
            library.add_publication(new_book)
            sg.popup("New Book Added Successfully.") #แสดงข้อความแจ้งเตือนว่าบันทึกข้อมูลหนังสือใหม่
สำเร็จ
            break #ออกจาก loop และจบการทำงานของโปรแกรม
    window.close() #ปิดหน้าต่าง GUI หลังจากเสร็จสิ้นการทำงาน


# ฟังก์ชันสำหรับค้นหาหนังสือ
def search_book_gui(library): #กำหนดโครงร่างของหน้าต่าง GUI สำหรับค้นหาหนังสือ
    layout = [
        [sg.Text('Enter book title: '), sg.InputText(key='-TITLE-')],
        [sg.Button('Search'), sg.Button('Cancel')]
    ]
    window = sg.Window('Search Book', layout) #สร้างหน้าต่าง GUI สำหรับค้นหาหนังสือ
    while True:
        event, values = window.read() #รอรับเหตุการณ์และข้อมูลที่ผู้ใช้กรอกเข้ามาในหน้าต่าง GUI
        if event == sg.WINDOW_CLOSED or event == 'Cancel':
            break #ออกจาก loop และจบการทำงานของโปรแกรม
        elif event == 'Search':
            title = values['-TITLE-']
            book = library.find_publication(title)
            if book:
                sg.popup(f"Title: {book.title}\nAuthor: {book.author}\nYear:
{book.year}\nISBN: {book.isbn}\nType: {book.book_type}")
            else:
                sg.popup("Book not found.")
            break #ออกจาก loop และจบการทำงานของโปรแกรม
    window.close() #ปิดหน้าต่าง GUI หลังจากเสร็จสิ้นการทำงาน

# ฟังก์ชันสำหรับยืมหนังสือ
def borrow_book_gui(library): #กำหนดโครงร่างของหน้าต่าง GUI สำหรับยืมหนังสือ
    layout = [ # สร้างหน้าต่าง GUI โดยใช้โครงร่าง layout ที่กำหนด
        [sg.Text('Enter member ID: '), sg.InputText(key='-ID-')],
        [sg.Text('Enter book title to borrow: '), sg.InputText(key='-TITLE-')],
        [sg.Button('Borrow Book'), sg.Button('Cancel')]
    ]
    window = sg.Window('Borrow Book', layout)
```

```
    while True:
        event, values = window.read() #รอรับเหตุการณ์และข้อมูลที่ผู้ใช้กรอกเข้ามาในหน้าต่าง GUI
        if event == sg.WINDOW_CLOSED or event == 'Cancel':
            break #ออกจาก loop และจบการทำงานของโปรแกรม
        elif event == 'Borrow Book':
            member_id = values['-ID-'] #ดึงข้อมูลหมายเลขสมาชิกที่ผู้ใช้กรอกเข้ามา
            member = library.find_member(member_id) #ค้นหาข้อมูลสมาชิกในระบบโดยใช้เมธอด
find_member ในคลาส Library
            if not member: # หากไม่พบข้อมูลสมาชิก
                sg.popup("Member not found.") #แสดงข้อความแจ้งเตือนว่าไม่พบข้อมูลสมาชิก
                break #ออกจาก loop และจบการทำงานของโปรแกรม
            title = values['-TITLE-']
            book = library.find_publication(title) #ค้นหาข้อมูลหนังสือในระบบโดยใช้เมธอด
find_publication ในคลาส Library
            if not book:
                sg.popup("Book not found.")
                break # ออกจาก loop และจบการทำงานของโปรแกรม
            borrow_date = datetime.now().strftime('%Y-%m-%d') #กำหนดวันที่ยืมเป็นวันปัจจุบันและ
แปลงเป็นรูปแบบ 'YYYY-MM-DD'
            due_date = (datetime.now() + timedelta(days=14)).strftime('%Y-%m-%d')
#กำหนดวันที่กำหนดส่งคืนเป็น 14 วันหลังจากวันปัจจุบันและแปลงเป็นรูปแบบ 'YYYY-MM-DD'
            if library.borrow_publication(member, book, borrow_date, due_date): #
ทำการยืมหนังสือด้วยเมธอด borrow_publication ในคลาส Library และตรวจสอบว่าเป็นไปด้วยความสำเร็จหรือไม่
                sg.popup("Book borrowed successfully.")
            else:
                sg.popup("Book is not available for borrowing.")
            break #ออกจาก loop และจบการทำงานของโปรแกรม
    window.close()  #ปิดหน้าต่าง GUI หลังจากเสร็จสิ้นการทำงาน

# ฟังก์ชันสำหรับแสดงรายละเอียดการยืม
def display_loan_details_gui(library): #กำหนดโครงร่างของหน้าต่าง GUI สำหรับแสดงรายละเอียดการยืมหนังสือ
    layout = [
        [sg.Text('Enter member ID: '), sg.InputText(key='-ID-')],
        [sg.Button('Display Loan Details'), sg.Button('Cancel')]
    ]
    window = sg.Window('Display Loan Details', layout) #สร้างหน้าต่าง GUI โดยใช้โครงร่าง layout
ที่กำหนด
    while True:
        event, values = window.read() #รอรับเหตุการณ์และข้อมูลที่ผู้ใช้กรอกเข้ามาในหน้าต่าง GUI
        if event == sg.WINDOW_CLOSED or event == 'Cancel':
            break #ออกจาก loop และจบการทำงานของโปรแกรม
        elif event == 'Display Loan Details':
            member_id = values['-ID-'] #ดึงข้อมูลหมายเลขสมาชิกที่ผู้ใช้กรอกเข้ามา
            member = library.find_member(member_id) #ค้นหาข้อมูลสมาชิกในระบบโดยใช้เมธอด
find_member ในคลาส Library
```

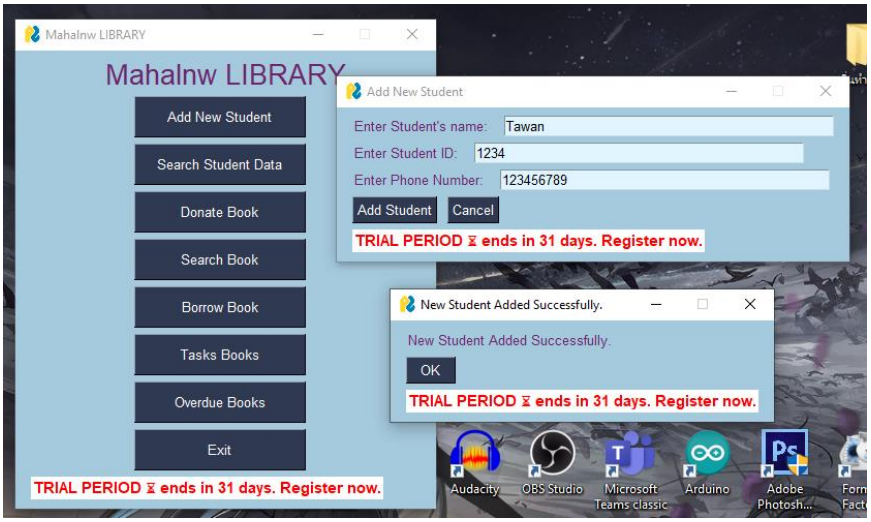
```
            if member: #ค้นหารายการยืมหนังสือทั้งหมดของสมาชิก
                loans = [loan for loan in library.loans if loan.member == member]
                if loans: #สร้างข้อความรายละเอียดการยืมหนังสือ
                    loan_details = '\n'.join([f"Publication:
{loan.publication.title}\nBorrow Date: {loan.borrow_date}\nDue Date:
{loan.due_date}" for loan in loans])
                    sg.popup("Loan Details:", loan_details) #แสดงข้อความรายละเอียดการยืมหนังสือ
                else:
                    sg.popup("No loans found for this member.") #แสดงข้อความว่าไม่พบการยืม
หนังสือสำหรับสมาชิกนี้
            else:
                sg.popup("Member not found.") #แสดงข้อความแจ้งเตือนว่าไม่พบข้อมูลสมาชิก
            break #ออกจาก loop และจบการทำงานของโปรแกรม
    window.close() #ปิดหน้าต่าง GUI หลังจากเสร็จสิ้นการทำงาน

# ฟังก์ชันสำหรับแสดงหนังสือที่เลยกำหนดคืน
def display_overdue_books_gui(library): #ค้นหาหนังสือที่มีกำหนดส่งคืนผ่านวันที่ปัจจุบัน
    overdue_books = [loan.publication for loan in library.loans if
datetime.strptime(loan.due_date, '%Y-%m-%d') < datetime.now()]
    if overdue_books: #สร้างรายการหนังสือที่เกินกำหนดส่งคืนเป็นข้อความ
        book_list = '\n'.join([f"- {book.title} by {book.author}" for book in
overdue_books])
        sg.popup("Overdue Books:", book_list) #แสดงข้อความรายการหนังสือที่เกินกำหนดส่งคืน
    else:
        sg.popup("No overdue books found.") #แสดงข้อความว่าไม่มีหนังสือที่เกินกำหนดส่งคืน

# ฟังก์ชันสำหรับระบบหลัก
def main():
    sg.theme('LightBlue1') #กำหนดธีมของ GUI

    layout = [
    [sg.Text('Mahalnw LIBRARY', font=('Helvetica', 20),pad=((75, 0)))],
    [sg.Column(layout=[[sg.Button('Add New Student', size=(20, 2),pad=((100,
0)))]])],
    [sg.Column(layout=[[sg.Button('Search Student Data', size=(20, 2),pad=((100,
0)))]])],
    [sg.Column(layout=[[sg.Button('Donate Book', size=(20, 2),pad=((100,
0)))]])],
    [sg.Column(layout=[[sg.Button('Search Book', size=(20, 2),pad=((100,
0)))]])],
    [sg.Column(layout=[[sg.Button('Borrow Book', size=(20, 2),pad=((100,
0)))]])],
    [sg.Column(layout=[[sg.Button('Tasks Books', size=(20, 2),pad=((100,
0)))]])],
```

```
    [sg.Column(layout=[[sg.Button('Overdue Books', size=(20, 2),pad=((100,
0)))]])],
    [sg.Column(layout=[[sg.Button('Exit', size=(20, 2),pad=((100, 0)))]])],
]



    window = sg.Window('Mahalnw LIBRARY', layout) #สร้างหน้าต่าง GUI ของโปรแกรมที่มีชื่อว่า "Mahalnw
LIBRARY"

    library = Library() #สร้างอ็อบเจกต์ของคลาส Library

    while True: #รอรับเหตุการณ์จากผู้ใช้ผ่านหน้าต่าง GUI
        event, _ = window.read()

        if event == sg.WINDOW_CLOSED or event == 'Exit':  #ถ้าปุ่มปิดหน้าต่างถูกคลิกหรือผู้ใช้ปิดหน้าต่าง
GUI
            break
        elif event == 'Add New Student':
            add_new_member_gui(library) # เรียกใช้ฟังก์ชันเพื่อเพิ่มสมาชิกใหม่
        elif event == 'Search Student Data':
            search_member_gui(library)  #เรียกใช้ฟังก์ชันเพื่อผู้ใช้ค้นหาข้อมูลสมาชิกในห้องสมุด
        elif event == 'Donate Book':
            add_new_book_gui(library) #เรียกใช้ฟังก์ชันเพื่อให้ผู้ใช้บริจาคหนังสือใหม่เข้าสู่ห้องสมุด
        elif event == 'Search Book':
            search_book_gui(library) #เรียกใช้ฟังก์ชันเพื่อให้ผู้ใช้ค้นหาข้อมูลหนังสือในห้องสมุด
        elif event == 'Borrow Book':
            borrow_book_gui(library) #เรียกใช้ฟังก์ชันเพื่อให้ผู้ใช้ยืมหนังสือจากห้องสมุด
        elif event == 'Tasks Books':
            display_loan_details_gui(library)  #เรียกใช้ฟังก์ชันเพื่อแสดงรายละเอียดของการยืมหนังสือทั้งหมด
        elif event == 'Overdue Books':
            display_overdue_books_gui(library) #เรียกใช้ฟังก์ชันเพื่อแสดงรายชื่อหนังสือที่เกินกำหนดคืน
    window.close()

if __name__ == "__main__":
    main()
```

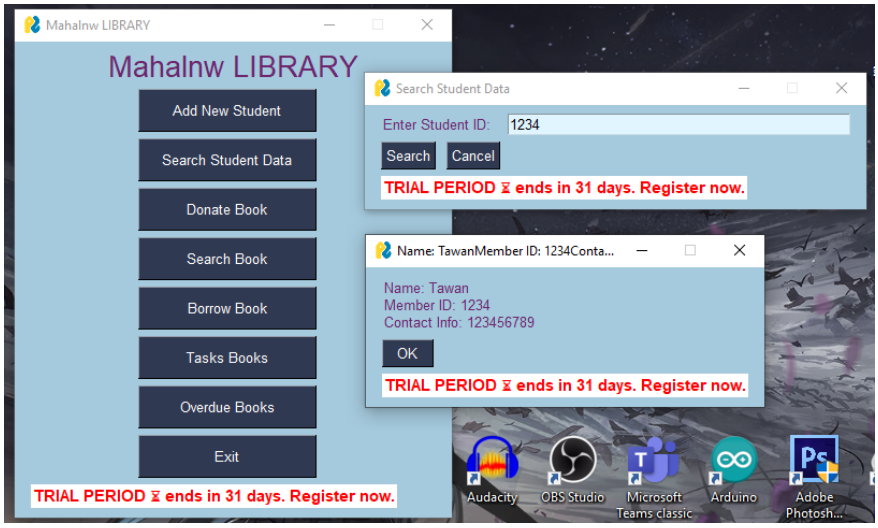
ผลการรัน

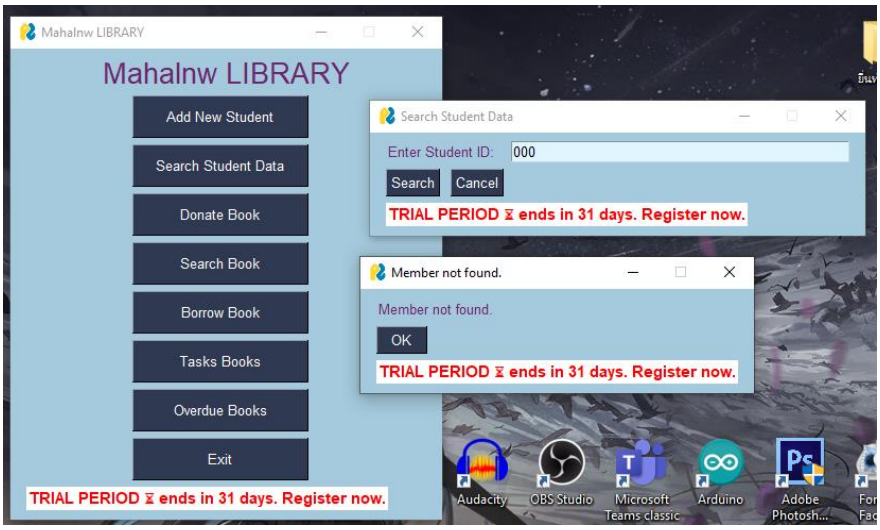
ส่วนของ PySimpleGUI

1) Add New Student

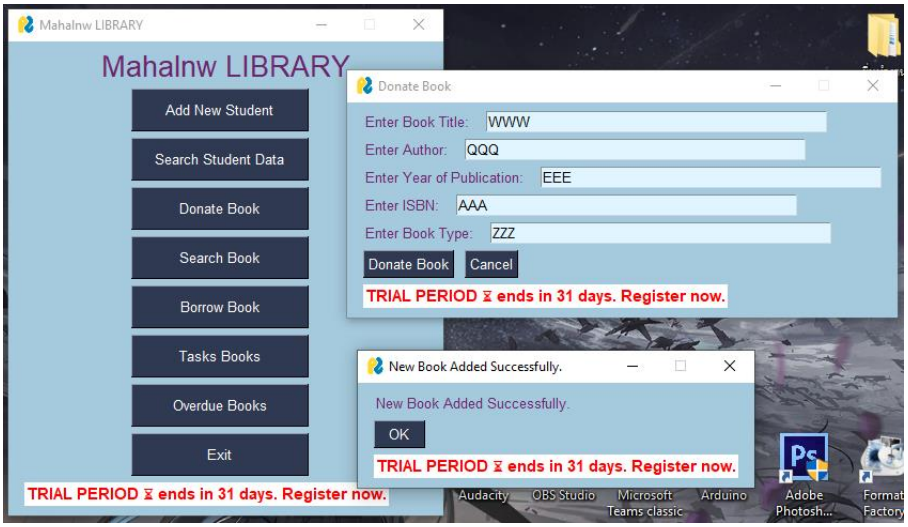


2) Search Student Data

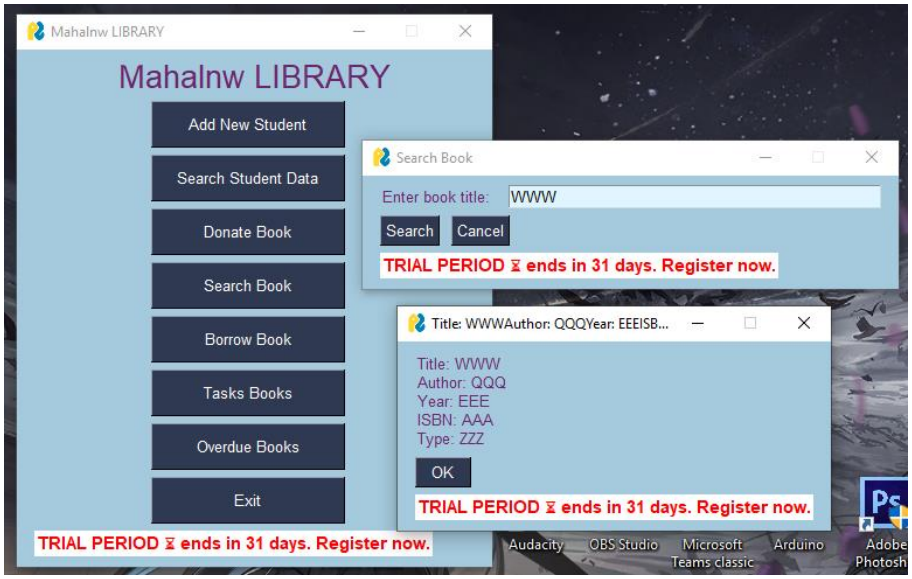




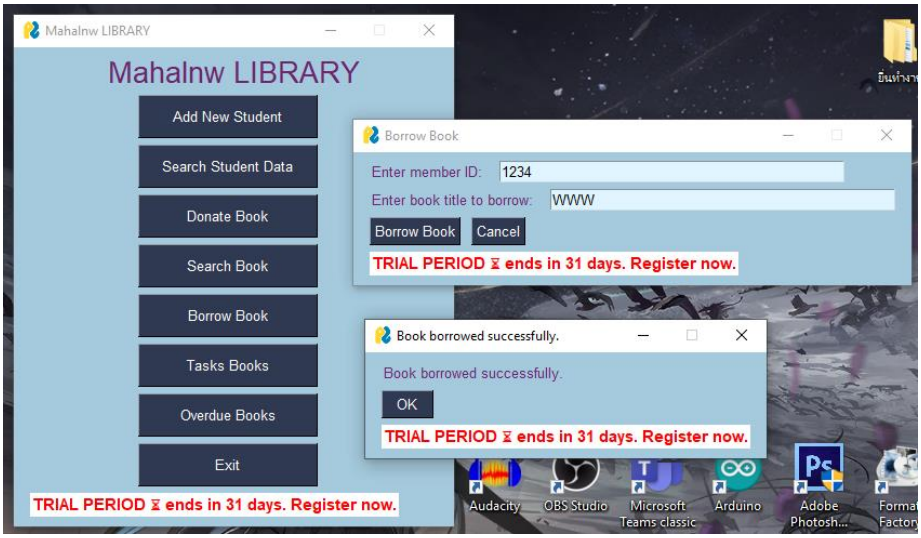
3) Donate Book



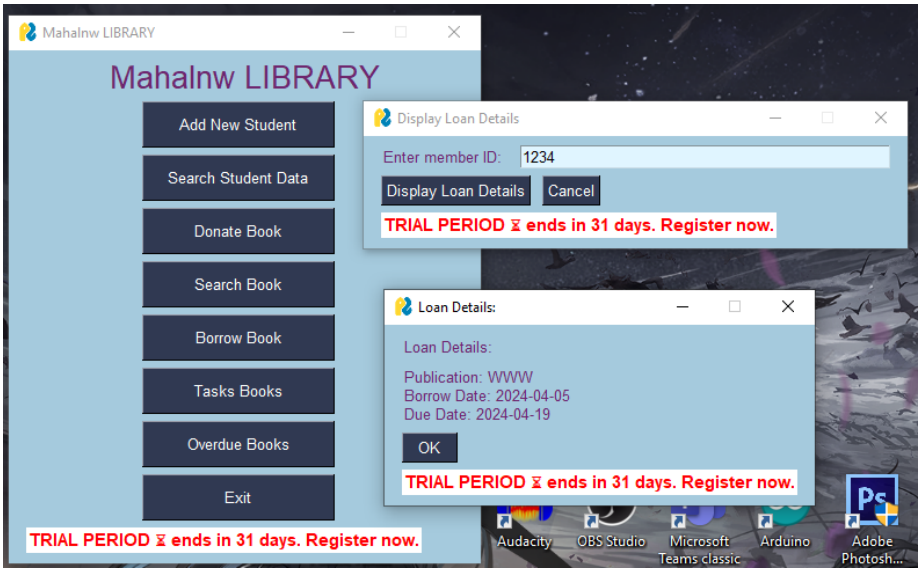
4) Search Book



5) Borrow Book



6) Tasks Book



7) Overdue Book