

Project Name: Project 2 Part 3 (Library Management System (LMS))

Contributors: Montgomery Benjamin, Moreland Nicolas, Safat Saniah

HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence. I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Task 1:

Query_1:

```
ALTER TABLE book_loans
ADD COLUMN Late INT;

UPDATE book_loans
SET Late = CASE
    WHEN date_returned > due_date THEN 1
    ELSE 0
END;
```

Query_2:

```
ALTER TABLE library_branch
ADD COLUMN LateFee DECIMAL(5,2);

UPDATE library_branch
SET LateFee = CASE
    WHEN Branch_ID = 1 THEN 0.50
    WHEN Branch_ID = 2 THEN 0.75
    WHEN Branch_ID = 3 THEN 1.00
    ELSE 0
END;
```

Create View

```
CREATE VIEW IF NOT EXISTS vBookLoanInfo AS
SELECT
    Book_Loans.card_no,
    Borrower.name,
    Book_Loans.date_out,
    Book_Loans.due_date,
    Book_Loans.date_returned,
    (julianday(Book_Loans.date_returned) - julianday(Book_Loans.date_out)) * 7 AS
    TotalDays,
    Book.title,
    CASE
        WHEN Book_Loans.date_returned > Book_Loans.due_date
            THEN (julianday(Book_Loans.date_returned) - julianday(Book_Loans.due_date))
        ELSE 0
    END
```

```
END AS DaysLate,
Book_Loans.branch_id,
CASE
    WHEN Book_Loans.date_returned > Book_Loans.due_date
        THEN ((julianday(Book_Loans.date_returned) - julianday(Book_Loans.due_date)))
    * Library_Branch.LateFee)
    ELSE 0
END AS LateFeeBalance
FROM
    Book_Loans
    JOIN Borrower ON Book_Loans.card_no = Borrower.card_no
    JOIN Book ON Book_Loans.book_id = Book.book_id
    JOIN Library_Branch ON Book_Loans.branch_id = Library_Branch.branch_id
ORDER BY
    Book_Loans.card_no ASC,
    Borrower.name ASC,
    Book_Loans.date_out ASC;
```

Select View

```
SELECT * FROM vBookLoanInfo;
```

Result Grid

Book Loan Info									
Card_No	Borrower Name	Date_Out	Due_Date	Returned_date	TotalDays	Book Title	Days_Later_Return	Branch ID	LateFeeBalance
111111	Alex Kim	2022-01-09	2022-02-09	2022-02-06	28.0	Brave New World	0	1	0
121212	Chloe Park	2022-01-18	2022-02-18	2022-02-18	31.0	The Da Vinci Code	0	3	0
123456	John Smith	2022-01-01	2022-02-01	2022-02-01	31.0	To Kill a Mockingbird	0	1	0
222222	Rachel Lee	2022-01-10	2022-02-10	2022-02-07	28.0	The Picture of Dorian Gray	0	2	0
232323	William Chen	2022-03-24	2022-03-31	2022-03-31	7.0	The Adventures of Huckleberry	0	1	0
234567	Emily Lee	2022-01-06	2022-02-06	2022-02-10	35.0	Animal Farm	4.0	2	3.0
333333	William Johnson	2022-03-01	2022-03-08	2022-02-08	-21.0	The Alchemist	0	1	0
343434	Olivia Johnson	2022-01-21	2022-02-21	2022-02-21	31.0	The Adventures of Tom Sawyer	0	3	0
345678	Bob Johnson	2022-01-03	2022-02-03	2022-02-03	31.0	Pride and Prejudice	0	2	0
444444	Ethan Martinez	2022-03-03	2022-03-10	2022-03-10	7.0	The God of Small Things	0	3	0
454545	Dylan Kim	2022-01-24	2022-02-24	2022-02-24	31.0	A Tale of Two Cities	0	3	0
456789	Laura Chen	2022-01-08	2022-02-08	2022-03-10	61.0	Lord of the Flies	30.0	3	30.0
555555	Grace Hernandez	2022-02-03	2022-03-03	2022-02-18	15.0	Wuthering Heights	0	3	0
565656	Sophia Park	2022-01-14	2022-02-14	2022-03-31	76.0	The Hobbit	45.0	1	22.5
567890	Tom Lee	2022-01-05	2022-02-05	2022-02-09	35.0	One Hundred Years of Solitude	4.0	1	2.0
676767	Olivia Lee	2022-01-15	2022-02-15	2022-02-21	37.0	The Lord of the Rings	6.0	3	6.0
787878	Noah Thompson	2022-03-05	2022-03-12	2022-02-24	-9.0	The Hitchhiker's Guide to the G	0	2	0
789012	Jane Doe	2022-01-02	2022-02-02	2022-02-02	31.0	1984	0	1	0
890123	Michael Park	2022-01-07	2022-02-07	2022-03-08	60.0	The Catcher in the Rye	29.0	2	21.75
901234	Sarah Kim	2022-01-04	2022-02-04	2022-02-04	31.0	The Great Gatsby	0	3	0
989898	Olivia Smith	2022-03-23	2022-03-30	2022-03-30	7.0	The Diary of a Young Girl	0	3	0

Action Output Response

There were a total of twenty-one (21) rows returned!

Task 2:

Sub_Task 1:

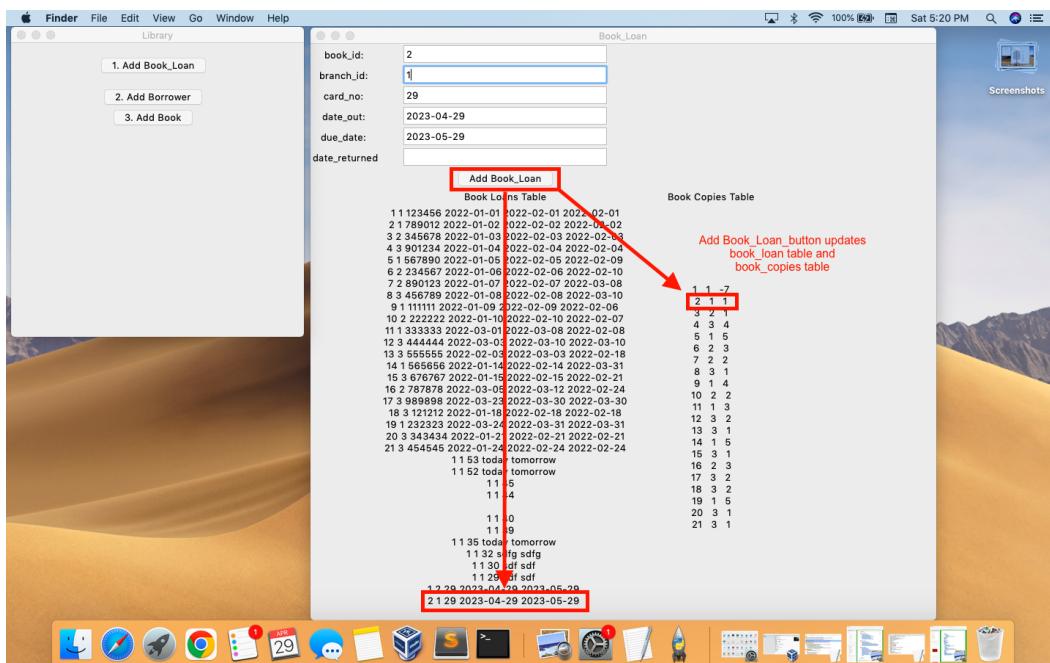
#add book

```
submit_cur.execute("INSERT INTO book_loans VALUES(:book_id, :branch_id,
:card_no, :date_out, :due_date, :date_returned)",
{
    'book_id': book_id,
    'branch_id': branch_id,
    'card_no': card_no,
    'date_out': date_out,
    'due_date': due_date,
    'date_returned': date_returned
})
```

#update book_copies table

```
submit_cur.execute("UPDATE Book_Copies SET No_Of_Copies = No_Of_Copies - 1
WHERE book_id = ? AND branch_id = ?", (book_id, branch_id))
```

#screenshot

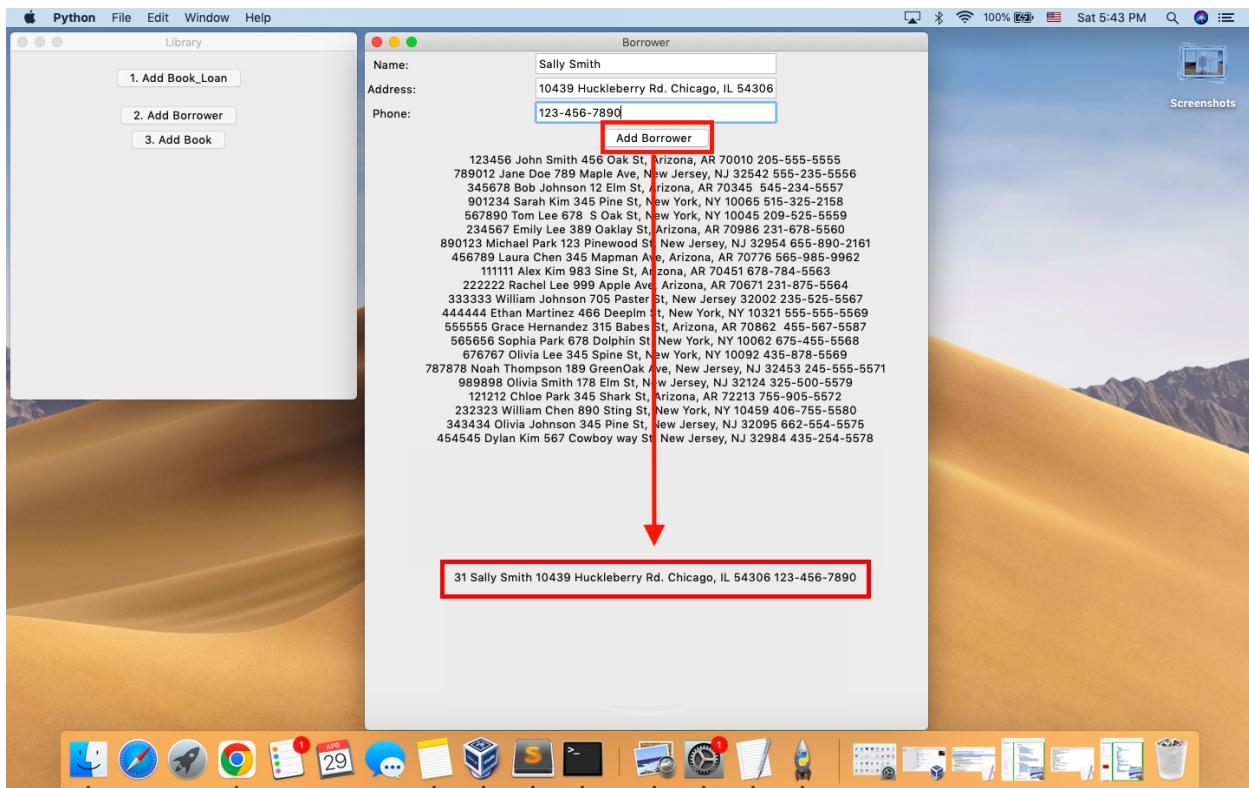


Sub_Task 2:

#add borrower

```
submit_cur.execute("INSERT INTO Borrower VALUES(:card_no, :name, :address,  
:phone)",  
{  
    'card_no': submit_cur.execute("SELECT count(*) from Borrower").fetchone()[0] +  
    1,  
    'name': name,  
    'address': address,  
    'phone': phone  
})
```

#screenshot



#description:

Per Task_2.2 requirements, borrower_id is not taken as input. It is calculated, uniquely (borrower_id = num_borrowers + 1)

Sub_Task 3:

#Add Book

```
submit_cur.execute("INSERT INTO book (book_id, title, publisher_name) VALUES (:book_id, :title, :publisher_name) ",  
{  
    'book_id': book_id,  
    'title': title,  
    'publisher_name': publisher_name  
})
```

```
submit_cur.execute("INSERT INTO BOOK_AUTHORS (Book_Id, Author_Name)  
VALUES (:Book_Id, :Author_Name) ",  
{  
    'Book_Id': book_id,  
    'Author_Name': author_name  
})
```

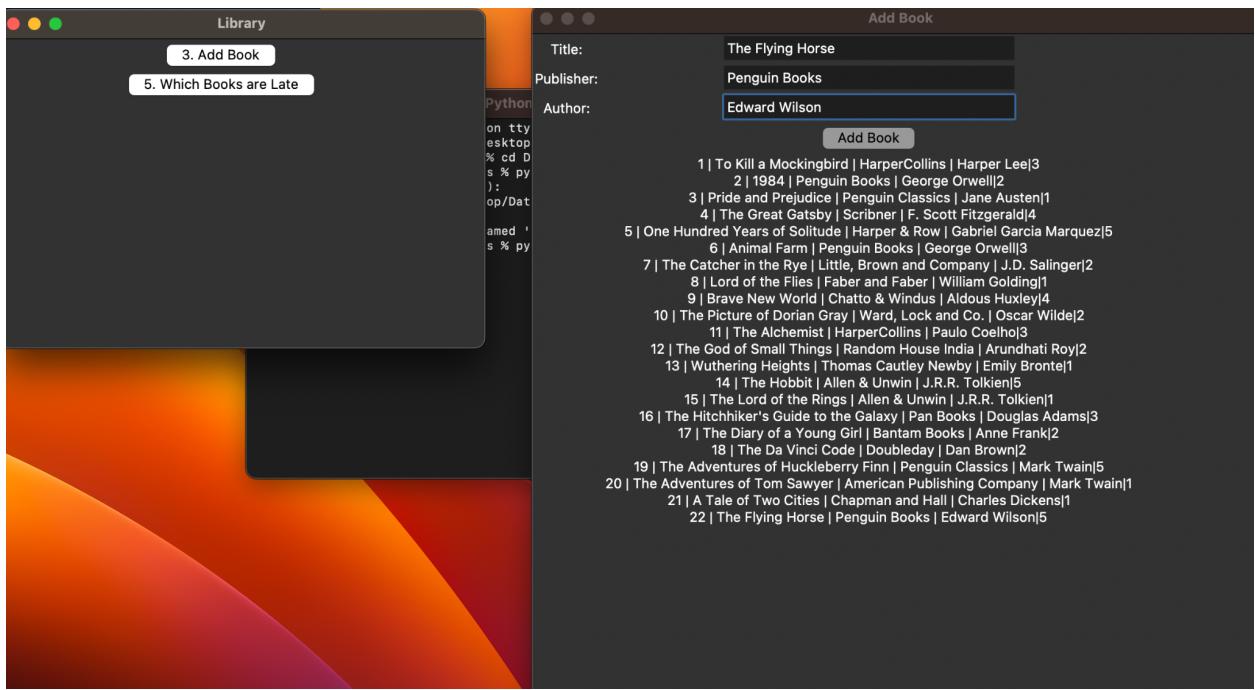
Query branch_id values from LIBRARY_BRANCH table

```
submit_cur.execute("SELECT branch_id FROM library_branch")  
branch_ids = submit_cur.fetchall()
```

Loop over all book_id and branch_id combinations, and insert No_OfCopies = 5

```
for branch_id_as_tuple in branch_ids:  
    branch_id, = branch_id_as_tuple  
    submit_cur.execute("INSERT INTO BOOK_COPIES (Book_Id, Branch_Id,  
    No_OfCopies) VALUES (:book_id, :branch_id, :no_of_copies)",  
    {  
        'book_id' : str(book_id),  
        'branch_id' : str(branch_id),  
        'no_of_copies' : '5'  
    })
```

#screenshot



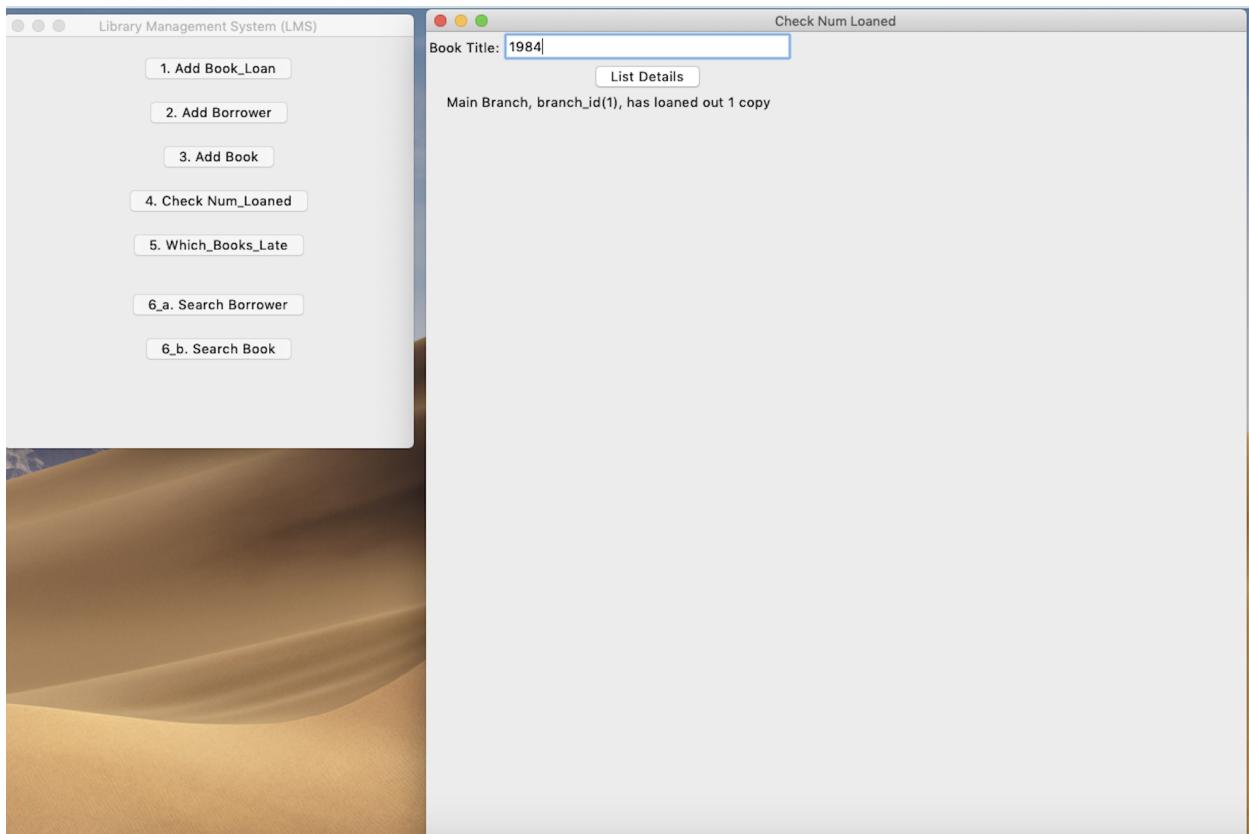
Description:

Adding a book title, publisher name and author name to the list of books.

Sub_Task 4: (...)

```
#book_loans
submit_cur.execute("SELECT library_branch.branch_name, library_branch.branch_id,
COUNT(*)
FROM (library_branch NATURAL JOIN book_loans) NATURAL JOIN book_copies
NATURAL JOIN book WHERE book.title = ? GROUP BY library_branch.branch_id",
(book_title,))
```

#Screenshot



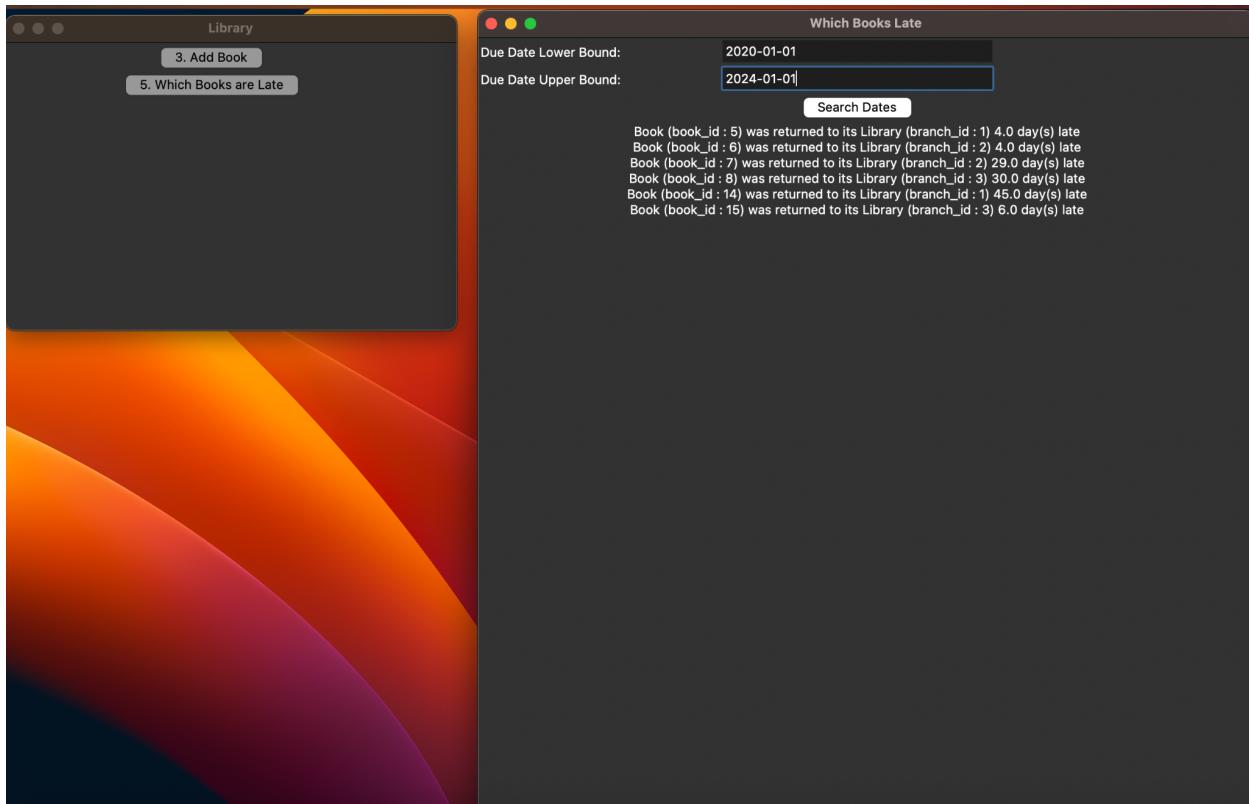
Description: Given any book title, retrieving the branch id, branch title and the number of copies loaned out.

Sub_Task 5:

```
#which_books_late
```

```
submit_cur.execute(""" SELECT book_id, branch_id, strftime('%J', date_returned) -  
strftime('%J', due_date) FROM book_loans WHERE date_returned > due_date  
AND due_date >= :due_date_lower_bound  
AND due_date <= :due_date_upper_bound  
ORDER BY due_date ASC """ , {  
    'due_date_lower_bound': due_date_lower_bound,  
    'due_date_upper_bound': due_date_upper_bound  
})
```

```
#screenshot
```



Description:

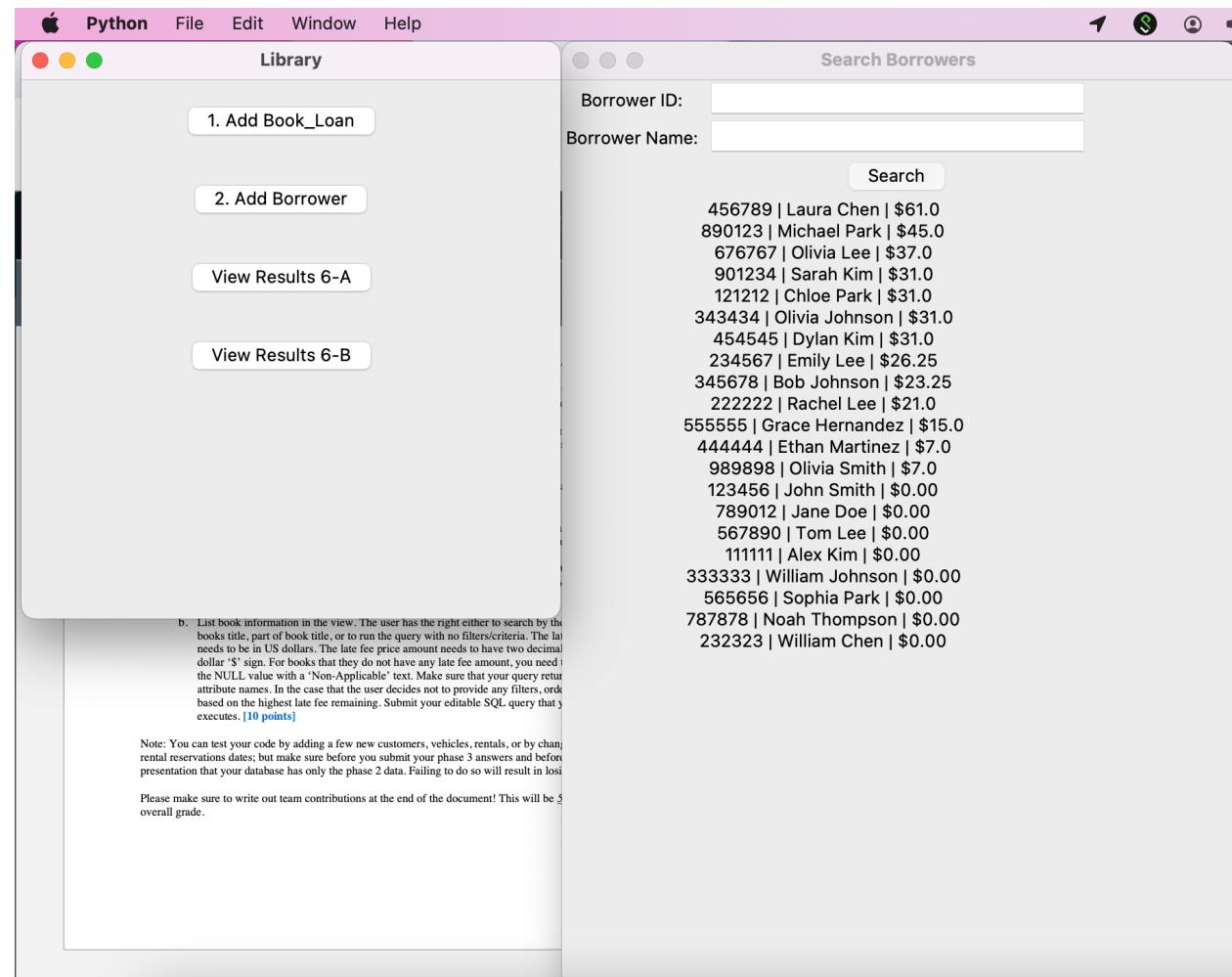
Given a range of dates, it returns how many books are late in that range and how many days it is late.

Sub_Task 6a: (...)

search borrower

```
query = "SELECT b.card_no AS 'Borrower ID', b.name AS 'Borrower Name', CASE  
WHEN bl.date_returned > bl.due_date THEN ('$' ||  
CAST(ROUND((julianday(bl.date_returned) - julianday(bl.due_date)) * lb.LateFee,  
2) AS DECIMAL(10,2))) ELSE '$0.00' END AS 'Late Fee' FROM Book_Loans bl JOIN  
Borrower b ON bl.card_no = b.card_no JOIN LIBRARY_BRANCH lb ON bl.branch_id =  
lb.branch_id WHERE 1=1 "  
  
if borrower_id:  
    query += f"AND b.card_no = {borrower_id} "  
if borrower_name:  
    query += f"AND b.name LIKE '%{borrower_name}%' "  
  
query += " ORDER BY CASE WHEN bl.date_returned > bl.due_date THEN  
(julianday(bl.date_returned) - julianday(bl.due_date)) * lb.LateFee ELSE NULL  
END DESC;"
```

screenshot



Sub_Task 6b: (...)

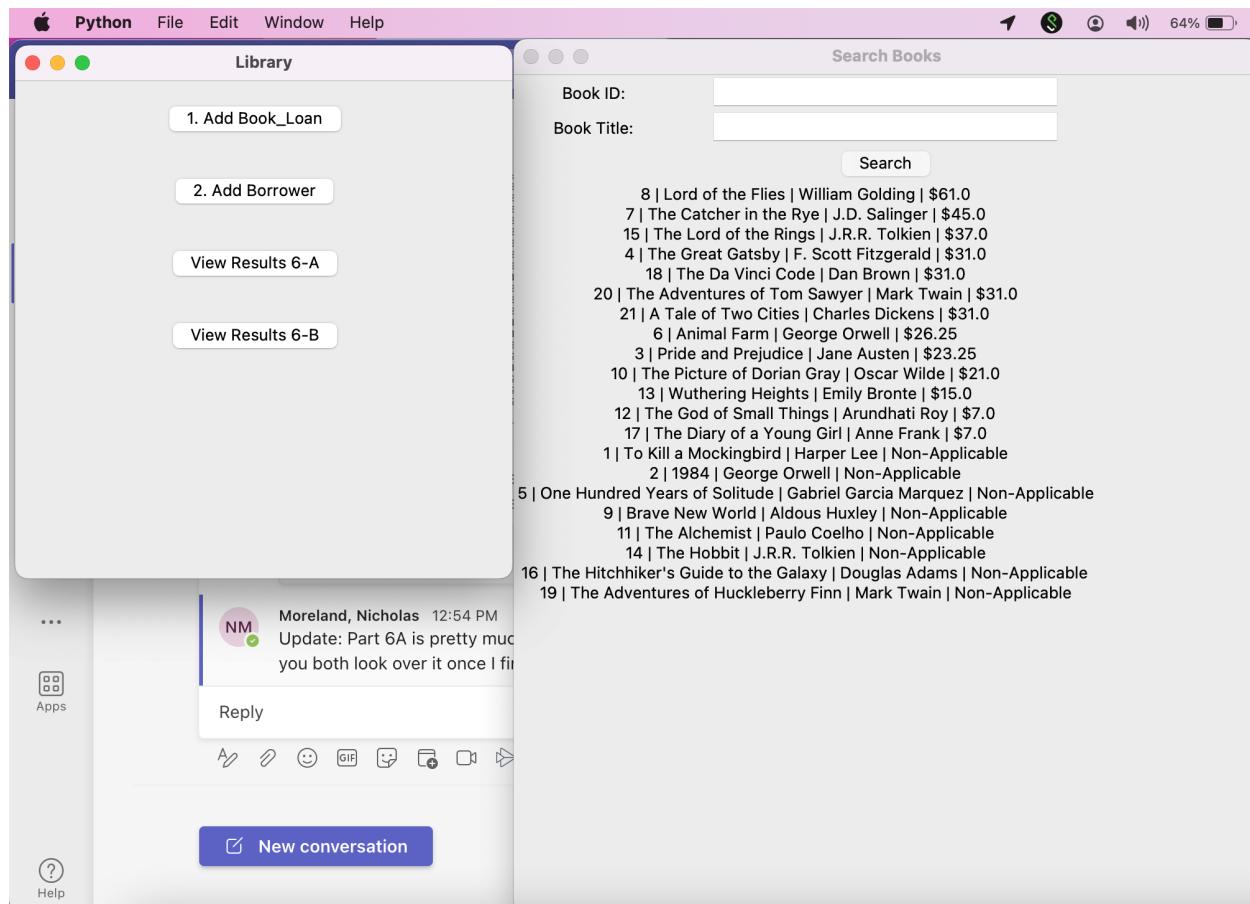
search book

```
query = "SELECT b.book_id AS 'Book ID', b.title AS 'Book Title', ba.Author_Name AS 'Author', CASE WHEN bl.date_returned > bl.due_date THEN ('$' || CAST(ROUND((julianday(bl.date_returned) - julianday(bl.due_date)) * lb.LateFee, 2) AS DECIMAL(10,2))) ELSE 'Non-Applicable' END AS 'Late Fee' FROM book b JOIN Book_Authors ba ON b.book_id = ba.book_id LEFT JOIN Book_Loops bl ON b.book_id = bl.book_id JOIN LIBRARY_BRANCH lb ON bl.branch_id = lb.branch_id WHERE 1=1"

if book_id:
    query += f" AND b.book_id = {book_id}"
if book_title:
    query += f" AND b.title LIKE '%{book_title}%'"

query += " ORDER BY CASE WHEN bl.date_returned > bl.due_date THEN (julianday(bl.date_returned) - julianday(bl.due_date)) * lb.LateFee ELSE NULL END DESC;"
```

screenshot



Contributions

Task_1: Nicolas Moreland

Task_2:

Sub_tasks 1–2: Montgomery Benjamin

Sub_tasks 3–5: Safat Saniah

Sub_tasks 6 (a–b): Moreland Nicolas

README.docx: Montgomery Benjamin

Report.pdf: Montgomery Benjamin, Moreland Nicolas, Safat Saniah

LMS.sql: Montgomery Benjamin