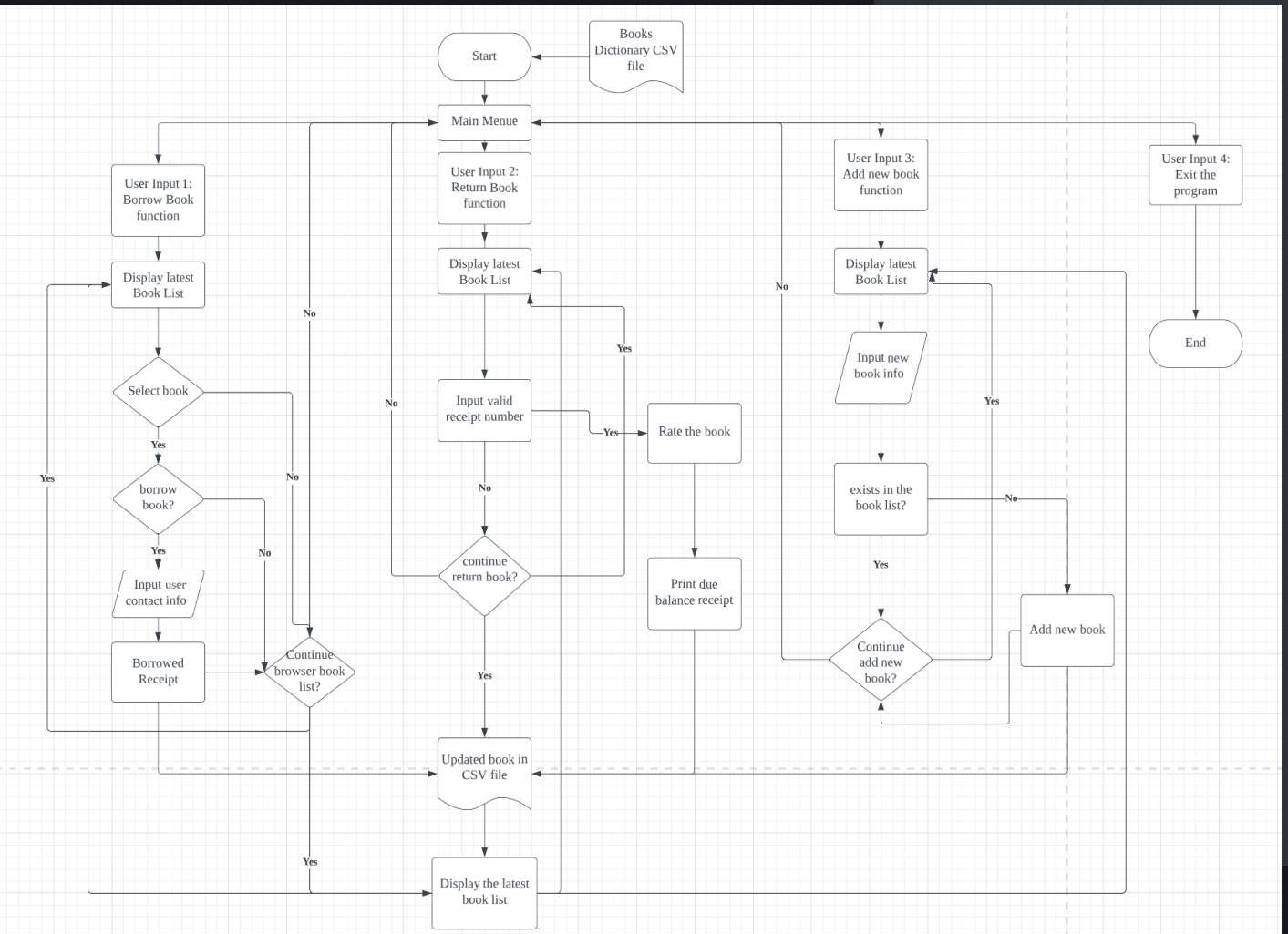
{Online Book Rental Store

T1A3 - Terminal Application

< Luying Han >

}



{Structure}

Online Book Rental Store app is a convenient terminal application that allows customers to repeatedly view the book listing, rent books, return books, and add books after starting to use the application.

Main Menu:

1. Borrow a book
2. Return a book
3. Add a new book
4. Exit the program

Main Menu

Display a menu of options to the user, and prompts them to enter a number corresponding to their desired action

Prompt_yes_no function used to control if user would like continue review the book list

Yes- continue the loop
No – exit browsing and return to main menu

```
# obtain user input1
if user_choice == "1":
    # def execute_user_choice_1(books):
    while True:
        display_books(books)
        selected_book = select_book(books)
        # if client select and confirm to borrow o book, update the book list
        if selected_book is not None:
            if borrow_book(selected_book) is not None:
                write_db(books, csv_file)
        if not prompt_yes_or_no(f"\n{fg(117)}{attr('bold')}Do you want to continue to browse our book list? (y/n): {attr('reset')}"):
            print(f"\n{fg(216)}{attr('bold')}{attr('bold')}Thank you for using our online borrow book service. {attr('reset')}")
            break
You, last week • bug fix finalize …
# obtain user input2
elif user_choice == "2":
    while True:
        display_books(books)
        returned_book = return_book(books)
        if returned_book is not None:
            write_db(books, csv_file)
            print(f"{fg(229)}\nUpdated book list: {attr(0)}")
            display_books(books)
            print(f"{fg(216)}{attr('bold')}\nThank you for using our online book return service. {attr(0)}")
            break
        else:
            if not prompt_yes_or_no(f"{fg(117)}{attr('bold')}\nDo you want to continue to return your book? (y/n):  {attr(0)}"):
                break

# obtain user input3
elif user_choice == "3":
    while True:
        display_books(books)
        add_book(books)
        write_db(books, csv_file)
        display_books(books)
        if not prompt_yes_or_no(f"\n{fg(193)}{attr('bold')}Do you want to continue to add new book? (y/n):  {attr(0)}"):
            print(f"\n{fg(216)}{attr('bold')}Thank you for using our online adding book service. {attr(0)}")
            break

elif user_choice == "4":
    write_db(books, csv_file)
    print(f"{fg(216)}{attr('bold')}\nThank you for using our online book rental store. See you next time! {attr(0)}")
    sys.exit()
```

Main Menu

```python
def prompt_yes_or_no(prompt):
    while True:
        confirm_browse = input(prompt).lower()
        if confirm_browse not in ["y", "n"]:
            print(f"\n{fg(229)}Sorry, the option you have entered is not valid, please enter 'y' or 'n'. {attr(0)}")
        else:
            return confirm_browse == "y"      You, 4 days ago • features.md
```

```
Welcome to our online book rental store. Please choose your service:

1. Borrow a book
2. Return a book
3. Add a wish list book
4. Exit the program

Please enter your choice:
```

Borrow Book Feature

This feature allows users to iterate borrow a book online.

Display book function

It imports the 'PrettyTable' module to create a well-organized and easily readable table to present the book information stored in a CSV document to customers.
It first defines the column headers. Next, the program iterates through a list of books, appending the data of each book as a row to the table. Finally, the program prints out the entire table containing all book details, making it simple for customers to comprehend the availability of each book.

LUYINGHAN_T1A3 > src > db.csv

You, 5 hours ago | 1 author (You)

```
id,name,author,rental_price,status,due_date,book_rate,receipt_number
001,Python Crash Course,Eric Matthes,17.9,available,None,4.5,0
002,Web Scraping with Python,Ryan Mitchell,19.0,unavailable,2023-05-05,3.6,44
003,Python Data Science Handbook,Jake VanderPlas,22.0,available,None,3.6,0
004,Expert Python Programming,Tarek Ziade,15.7,unavailable,2023-05-14,3.0,10
005,Python Network Programming,Dr. M. O. Faruque Sarker,23.5,available,None,4.2,0
006,I love python,Helena.Han,0.0,unavailable,unavailable,0.0,0
```

```python
def display_books(books):
    # create a table to display book list
    table = PrettyTable(["ID", "Name", "Author", "Rental Price",
                         "Status", "Due Date", "Book Rate", "Receipt Number"])
    for book in books:
        row = [book["id"], book["name"], book["author"], book["rental_price"],
               book["status"], book["due_date"], book["book_rate"], book["receipt_number"]]
        table.add_row(row)
        for row in table._rows:
            for i, cell in enumerate(row):
                if cell == 0:
                    row[i] = "N/A"
    print(f"{fg('cyan')}{attr('bold')}\nHere is the list of books for rental: {attr('reset')}")
    print(table)
```

You, 4 days ago • create bash file

```
Welcome to our online book rental store. Please choose your service:

1. Borrow a book
2. Return a book
3. Add a wish list book
4. Exit the program

Please enter your choice: 1

Here is the list of books for rental:
```

ID	Name	Author	Rental Price	Status	Due Date	Book Rate	Receipt Number
001	Python Crash Course	Eric Matthes	17.9	available	None	4.5	N/A
002	Web Scraping with Python	Ryan Mitchell	19.0	unavailable	2023-05-05	3.6	44
003	Python Data Science Handbook	Jake VanderPlas	22.0	available	None	3.6	N/A
004	Expert Python Programming	Tarek Ziade	15.7	unavailable	2023-05-14	3.0	10
005	Python Network Programming	Dr. M. O. Faruque Sarker	23.5	available	None	4.2	N/A
006	I love python	Helena.Han	N/A	unavailable	unavailable	N/A	N/A

Borrow Book Feature

Select book function

1. Prompt user to input a 3-digital unique book id
2. List comprehension to create a new list that contains all books that matching the input id

```
# define a function for selected book
def select_book(books):
    # This function prompts the user to select a book from the given list of books
    # and returns the details of the selected book.
    while True:
        book_id = input(f"\n{fg(122)}Please enter the book ID you are interested: {attr(0)}")
        if not book_id.isdigit() or len(book_id) != 3:
            print(f"\n{fg(226)}Sorry, the book ID you have entered is not valid, please enter a valid 3-digit integer ID. {attr(0)}")
        else:
            break

    selected_book = [item for item in books if item["id"] == book_id]
    if len(selected_book) == 0:
        print(f"\n{fg(226)}Sorry, the book ID you have entered is not list in our online store. If you would like to add a new book, please press option 3. {attr(0)}")
        return

    selected_book = selected_book[0]
    if selected_book["status"] == "unavailable":
        if selected_book["due_date"] == "unavailable":
            print(f"\n{fg(226)}Sorry, the book will be add to our online store later, please check it after 7 days.")
        else:
            now = datetime.datetime.now()
            time_diff = datetime.datetime.strptime(
                selected_book["due_date"], "%Y-%m-%d").date() - now.date()
            print(f"\n{fg(226)}Sorry, the book is unavailable for rental currently. It will be available from {selected_book['due_date']}, {time_diff.days} days from today. {attr(0)}")
        return

    return selected_book
```

Borrow book function

Updates the information of select book and prints a receipt number for the transaction

Prompt user input their personal information using regex format

Generate unique receipt number

```python
def borrow_book(selected_book):
    # This function updates the information of the selected book and
    # print a receipt number for the transaction.

    if prompt_yes_or_no(f"\n{fg(229)}The book is currently available, do you want to borrow this book? (y/n): {attr(0)}"):

        print(f"\n{fg(122)}Please enter your personal information to complete the transaction.{attr(0)}")

        name = validate_name()
        address = validate_address()
        email = validate_email()
        phone = validate_phone()
        receipt_id = selected_book["id"]
        # create a receipt dictionary to store the receipt information
        now = datetime.datetime.now()

        receipt_num = generate_receipt_number()
        rental_price = selected_book["rental_price"]
        deposit = round(rental_price * 0.2, 2)

        receipt = {
            'receipt_num': receipt_num,
            "name": name,
            "address": address,
            "phone": phone,
            "email": email,
            "book_id": receipt_id,
            "book_name": selected_book["name"],
            "borrow_date": now.date(),
            "due_date": now.date() + datetime.timedelta(days=30),      You, now • Uncommitted changes
            "deposit": deposit,
        }
        print(f"\n{fg(216)}{attr('bold')}Thank you for borrowing {selected_book['name']}. Here is your receipt. {attr('reset')}")

        # create a table to display receipt information
        show_receipt(receipt)

        # update the selected book status and due date
        selected_book["status"] = "unavailable"
        selected_book["due_date"] = (now.date() + datetime.timedelta(days=30)).strftime("%Y-%m-%d")
        selected_book["receipt_number"] = receipt_num

        return selected_book
```

```
Please enter the book ID you are interested: 001

The book is currently available, do you want to borrow this book? (y/n): y

Please enter your personal information to complete the transaction.
Name: helena han
Address: jsfj@

Sorry, the address you have entered is not valid, please try again, format: 10-50 characters, only letters, numbers and space.
Address: 123 pitt street
Email: helena.han@gmail.com
Phone: 0291829887

Thank you for borrowing Python Crash Course. Here is your receipt.
+--------------------+----------------------+
| Receipt Number: 34 |     Information      |
+--------------------+----------------------+
|       Name:        |      helena han      |
|      Address:      |   123 pitt street    |
|       Phone:       |      0291829887      |
|       Email:       | helena.han@gmail.com |
|      Book ID:      |         001          |
|     Book Name:     | Python Crash Course  |
|    Borrow Date:    |      2023-05-01      |
|     Due Date:      |      2023-05-31      |
|      Deposit:      |         3.58         |
+--------------------+----------------------+

Do you want to continue to browse our book list? (y/n): y

Here is the list of books for rental:
+-----+----------------------------+----------------------+-------------+-------------+-------------+-----------+----------------+
|  ID |            Name            |        Author        | Rental Price |    Status   |   Due Date  | Book Rate | Receipt Number |
+-----+----------------------------+----------------------+-------------+-------------+-------------+-----------+----------------+
| 001 |     Python Crash Course    |     Eric Matthes     |     17.9    | unavailable | 2023-05-31  |    3.8    |       34       |
| 002 |   Web Scraping with Python |     Ryan Mitchell    |     19.0    |  available  |     None    |    4.2    |      N/A       |
| 003 | Python Data Science Handbook |    Jake VanderPlas   |     22.0    | unavailable | 2023-05-31  |    3.6    |       58       |
| 004 |  Expert Python Programming |      Tarek Ziade     |     15.7    | unavailable | 2023-05-28  |    4.0    |       72       |
| 005 |  Python Network Programming | Dr. M. O. Faruque Sarker |  23.5    |  available  |     None    |    4.2    |      N/A       |
| 006 |        I love python       |      Helena.Han      |     N/A     | unavailable | unavailable |    N/A    |      N/A       |
+-----+----------------------------+----------------------+-------------+-------------+-------------+-----------+----------------+

Please enter the book ID you are interested:
```

Return Book Feature

The function uses loops and input functions to get the receipt number for returning the book from the customer.

If the receipt number exists in the book list, the function updates the relevant book information

Prompts the customer to rate the book and updates the book's average rating based on the current rating value

Display the remaining balance table

```python
while True:
    # error handling for invalid input
    try:
        return_receipt_number = int(input(f"\n{fg(122)}Please enter your receipt number:  {attr(0)}"))
        if return_receipt_number <= 0:
            print(f"\n{fg(229)}Please enter a positive integer. {attr(0)}")
            continue
        # check if the receipt number is in the book list
        for book in books:
            if book["receipt_number"] == int(return_receipt_number):
                print(f"\n{fg(226)}Thank you for returning {book['name']}. {attr(0)}")
                while True:
                    # rate the book and update the book rate information
                    try:
                        current_book_rate = float(input(f"\n{fg(122)}Please rate the book you have borrowed:  {attr(0)}"))
                        if current_book_rate < 1 or current_book_rate > 5:
                            raise ValueError
                    except ValueError:
                        print(f"\n{fg(229)}Invalid input. Please enter a non-zero number (from 1-5)")
                    else:
                        average_rate = (book["book_rate"] + current_book_rate)/2

                        book["book_rate"] = float(format(average_rate, '.1f'))
                        book["status"] = "available"
                        book["due_date"] = "None"
                        book["receipt_number"] = 0

                        print(f"\n{fg(216)}{attr('bold')}Thank you for updating {book['name']}'s rate! {attr(0)}")
                        # deposit = rental price * 0.2
                        due_balance = float(book["rental_price"] -  book["rental_price"] * 0.2)
                        deposit = float(book["rental_price"] * 0.2)
                        print(f"{fg(229)}\nPlease pay your due balance: ${due_balance:.2f} {attr(0)}")
                        due_balance_table = PrettyTable(
                            ["Receipt Number", "Rental Price", "Deposit", "Due Balance"])
                        due_balance_table.add_row(
                            [return_receipt_number, book["rental_price"], f"{deposit:.2f}", "{:.2f}".format(due_balance)])
                        print(due_balance_table)
                        return book
        else:
            print(f"{fg(229)}\nThe receipt number you entered is not on the list. Please double check your receipt number. {attr(0)}")
            break
    except ValueError:
        print(f"{fg(229)}\nInvalid input. Please enter a valid non-zero integer. {attr(0)}")
```

You, 3 days ago • add feature done

Here is the list of books for rental:

ID	Name	Author	Rental Price	Status	Due Date	Book Rate	Receipt Number
001	Python Crash Course	Eric Matthes	17.9	unavailable	2023-05-31	3.8	34
002	Web Scraping with Python	Ryan Mitchell	19.0	available	None	4.2	N/A
003	Python Data Science Handbook	Jake VanderPlas	22.0	unavailable	2023-05-31	3.6	58
004	Expert Python Programming	Tarek Ziade	15.7	unavailable	2023-05-28	4.0	72
005	Python Network Programming	Dr. M. O. Faruque Sarker	23.5	available	None	4.2	N/A
006	I love python	Helena.Han	N/A	unavailable	unavailable	N/A	N/A

Please enter your receipt number: 58

Thank you for returning Python Data Science Handbook.

Please rate the book you have borrowed: 4.8

Thank you for updating Python Data Science Handbook's rate!

Please pay your due balance: $17.60

Receipt Number	Rental Price	Deposit	Due Balance
58	22.0	4.40	17.60

Updated book list:

Here is the list of books for rental:

ID	Name	Author	Rental Price	Status	Due Date	Book Rate	Receipt Number
001	Python Crash Course	Eric Matthes	17.9	unavailable	2023-05-31	3.8	34
002	Web Scraping with Python	Ryan Mitchell	19.0	available	None	4.2	N/A
003	Python Data Science Handbook	Jake VanderPlas	22.0	available	None	4.2	N/A
004	Expert Python Programming	Tarek Ziade	15.7	unavailable	2023-05-28	4.0	72
005	Python Network Programming	Dr. M. O. Faruque Sarker	23.5	available	None	4.2	N/A
006	I love python	Helena.Han	N/A	unavailable	unavailable	N/A	N/A

Thank you for using our online book return service.

1. Borrow a book
2. Return a book
3. Add a wish list book
4. Exit the program

Please enter your choice:

Add Book Feature

Function finds the maximum book ID from the book list.

It prompts the user to enter the information for the new book, including the book name and author name.

Uses a loop to check if the new book name already exists in the book list.

Exists – raise value error
Doesn't exists – add new book

```python
def add_book(books):
    print(f"\n{fg(122)}Please enter the following information to add a book: {attr(0)}")

    # add id from book list

    max_id = max(int(book["id"]) for book in books)

    try:
        book_name = validate_book_name()
        book_author = validate_author_name()
        for book in books:
            if book["name"].lower() == book_name.lower():
                raise ValueError(f"\nBook '{book_name}' already exists in the list.")

        # create a new book dictionary
        new_book = {}
        new_book["id"] = str(max_id + 1).zfill(3)
        new_book["name"] = book_name
        new_book["author"] = book_author
        new_book["rental_price"] = 0.0
        new_book["status"] = "unavailable"
        new_book["due_date"] = "unavailable"      You, yesterday • add test case
        new_book["book_rate"] = 0.0
        new_book["receipt_number"] = 0

        books.append(new_book)
        print(f"{fg(123)}{attr('bold')}\n{book_name} has been added to the list and will become available in 7 days.{attr(0)}")

    except ValueError as e:
        print(f"{fg(229)}{str(e)}{attr(0)}")

    return books
```

```
Thank you for using our online book return service.

1. Borrow a book
2. Return a book
3. Add a wish list book
4. Exit the program

Please enter your choice: 3

Here is the list of books for rental:
+-----+-----------------------------+-----------------------+--------------+-------------+------------+-------------+----------------+
|  ID |             Name            |         Author        | Rental Price |    Status   |  Due Date  | Book Rate | Receipt Number |
+-----+-----------------------------+-----------------------+--------------+-------------+------------+-------------+----------------+
| 001 |     Python Crash Course     |      Eric Matthes     |     17.9     | unavailable | 2023-05-31 |     3.8     |       34       |
| 002 |   Web Scraping with Python  |     Ryan Mitchell     |     19.0     |  available  |    None    |     4.2     |       N/A      |
| 003 | Python Data Science Handbook |    Jake VanderPlas    |     22.0     |  available  |    None    |     4.2     |       N/A      |
| 004 |  Expert Python Programming  |      Tarek Ziade      |     15.7     | unavailable | 2023-05-28 |     4.0     |       72       |
| 005 |  Python Network Programming | Dr. M. O. Faruque Sarker |     23.5     |  available  |    None    |     4.2     |       N/A      |
| 006 |       I love python         |       Helena.Han      |     N/A      | unavailable | unavailable |     N/A     |       N/A      |
+-----+-----------------------------+-----------------------+--------------+-------------+------------+-------------+----------------+

Please enter the following information to add a book:
Book Name: Python is Great
Book Author: Mike Lee

Python is Great has been added to the list and will become available in 7 days.

Here is the list of books for rental:
+-----+-----------------------------+-----------------------+--------------+-------------+------------+-------------+----------------+
|  ID |             Name            |         Author        | Rental Price |    Status   |  Due Date  | Book Rate | Receipt Number |
+-----+-----------------------------+-----------------------+--------------+-------------+------------+-------------+----------------+
| 001 |     Python Crash Course     |      Eric Matthes     |     17.9     | unavailable | 2023-05-31 |     3.8     |       34       |
| 002 |   Web Scraping with Python  |     Ryan Mitchell     |     19.0     |  available  |    None    |     4.2     |       N/A      |
| 003 | Python Data Science Handbook |    Jake VanderPlas    |     22.0     |  available  |    None    |     4.2     |       N/A      |
| 004 |  Expert Python Programming  |      Tarek Ziade      |     15.7     | unavailable | 2023-05-28 |     4.0     |       72       |
| 005 |  Python Network Programming | Dr. M. O. Faruque Sarker |     23.5     |  available  |    None    |     4.2     |       N/A      |
| 006 |       I love python         |       Helena.Han      |     N/A      | unavailable | unavailable |     N/A     |       N/A      |
| 007 |       Python is Great       |       Mike Lee        |     N/A      | unavailable | unavailable |     N/A     |       N/A      |
+-----+-----------------------------+-----------------------+--------------+-------------+------------+-------------+----------------+

Do you want to continue to add new book? (y/n):
```

Project Review

Challenges

> Time Management

> Testing and input Validation

Ethical issues

> Intellectual property

Favorite parts

> Prettytable

> prompt_yes_no function

Thank you for your attention