



Islington college
(इसलिंग्टन कलेज)

CS4051NI/CC4059NI-Fundamentals of Computing

60% Individual Coursework

1st Sit COURSEWORK

Summer Semester 2023

Student Name: Bibek Kumar Thakur

London Met ID: 22085816

College ID: NP01CP4S230060

Group: C19

Assignment Due Date: Friday, August 25, 2023

Assignment Submission Date: Friday, August 25, 2023

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Contents

1. Introduction:	2
1.1 Project Details:	2
1.2 Goals and Objectives:	2
1.3 Technologies used for the development of the project:	3
2. Discussion and Analysis	5
2.1 Algorithm:	5
2.2 Pseudocodes:	7
2.3 Flowchart:.....	14
2.4 Data Structures:	16
3. Testing of Program.	17
3.1 Test-1: Show the implementation of Try, Except.	17
3.2 Test-2: Selection rent and return of items.	18
3.3 Test-3: File generation of renting of items(s) (Renting multiple items(s):	20
3.4 Test-4: File generation of return process of items(s)	21
3.5 Test-5: Show the update in stock of items(s)	23
4. Conclusion:	26
4.1 Summary of project:.....	26
4.2 Development Process:	27
4.3 Things Learned:.....	28
4.4 Limitations and Challenges:	28
4.5 Solutions:	29
5.References	29
References	29
6. Appendix.....	30
6.1 Main.py Module.....	30
6.2 rental_functions.py Module:	31
6.3 display_functions.py Module	34
6.4 return_functions.py Module:	35
6.5 file_operations.py Module:	37

1. Introduction:

1.1 Project Details:

This Course work consist of a software program called the Event Equipment Rental Management System for an event equipment rental firm which can handle their inventory, rental transactions, and billing procedures more effectively. The system aims to automate equipment tracking, rental processing, and the creation of invoices for both rentals and returns.

1.2 Goals and Objectives: The Event Equipment Rental Management System's primary objectives are to increase the store's operational effectiveness, improve customer service, and assure reliable and transparent rental transactions. The system aims to attain the following goals:

- **Equipment Inventory Management :** Maintain a computerised inventory of the equipment that is available, including names, brands, rental costs, and stock levels. This facilitates quick access to equipment data and aids in accurate stock tracking.
- **Efficient Rental Processing:** Provide a user-friendly interface for renting equipment. Customers may choose the equipment they want and how much they want to rent. The system provides availability and decides the rental fee based on the cost of the equipment and the rental duration.
- **Automated Invoicing:** Generate customized invoices for each rental transaction. Information about the customer, rental dates, and the total amount due are all included in invoices. The system certain that the rental fees and any related fines are calculated accurately.
- **Timely Return Management:** Make it easier for clients to return borrowed equipment by allowing them for doing it. The system generates rental conditions, calculates fines for late returns, and creates bills in accordance with those conditions.

- **Stock Tracking and Updates:** Automatically update the stock quantities of rented and returned equipment. By avoiding both overbooking and understocking, clients are given correct availability information.
- **Minimise error:** By automating calculations and data entry, errors and mistakes in rental transactions may be reduced. This lowers error rates and provides accurate invoicing.
- **Customer Experience Improvement and Record Keeping:** Customers can trust the system to provide reliable rental and payment details. Maintain a digital record of all rental transactions and related invoices. This historical data can be useful for future reference, analysis, and reporting.

1.3 Technologies used for the development of the project:

- **IDLE(Python 3.10)** is Python's Integrated Development and Learning Environment.

IDLE has the following features:

- made entirely of Python code and the tkinter GUI toolkit.
- cross-platform: largely same behavior on Windows, Unix, and macOS Python
- The interactive Python shell window has colored input, output, and error messages.
- Multiple undo, Python colorizing, smart indent, call hints, auto completion, and more features are included in this multi-window text editor.
- search across numerous files (grep), alter text within editor windows, and search within any window.
- Debugging tool featuring stepping, permanent breakpoints, and namespace viewing.
- dialogues, browsers, and configuration (Python Software Foundation, 2001-2023).

➤ **Microsoft Word:**

The Report on this project was carried out in Microsoft Word. Microsoft Word is one of the most useful software for building text files and these text files can be sent from one creator to another and can be edited as well. It was first released in 1983. Any word file is by default saved with the extension “.doc”. This software includes grammar and spelling checker, font and text formatting, advance layout, image supporter and many more for professional reports, resumes, letters, etc (Geeksforgeeks, 2023).

- **Draw.io:** The flowchart of the program was made on draw.io. Draw.io is an online diagram editor built around google drive. Users may produce UML diagrams, flowcharts, network diagrams, entity relations, and more using draw.io. The fact that draw.io keeps the data on Google Drive is one of its benefits. There is no requirement for a further third party as a result. The export format is also complete, whether it is PNG, JPEG, SVG, PDF, XML, HTML, URL. You may pick from a variety of storage options with io, including your local device, Google Drive, OneDrive, Dropbox, and GitHub. A picture is worth a thousand words, so Draw.io can help you convey all those words with diagrams and flowcharts. Some specific examples of what businesses might use Draw.io for include:

- i. making a flowchart to depict a procedure or the project's workflow.
- ii. creating an organizational chart to display the corporate hierarchy or structure.
- iii. use a network diagram to visualize a network.
- iv. drawing up a website's or user interface's design.
- v. making a floor plan or layout diagram to demonstrate an idea (Paraschiv, 2023).

2. Discussion and Analysis

2.1 Algorithm:

A set of finite rules or instructions to be followed in calculations or other problem-solving operations is called an algorithm. Algorithms are used to automate processes or solve issues in a methodical and effective way. They are a set of guidelines or rules that direct software or computers in carrying out certain tasks or resolving issues (geeksforgeeks, 2023).

Algorithm of Program to Manage Equipment Rental Shop:

Step 1: Start the program.

Step 2: Import necessary tools.

Step 3: Define All Functions

a) "display_available_equipment(equipment_data)": #(This shows available equipment in shop like a list.)

b) "rent_equipment(equipment_data)": #(customer's name is asked. It holds record of the total money to be paid by customer. It Shows available equipment for the customers. Let the customer select what equipment/item they want to rent. If available, calculate the cost and confirm)

c) "return_equipment(equipment_data)": #(It Shows available equipment for the customers. Let the customer select what they want to return. Also, calculates any fine for late return and show total.)

Step 4: Define "main()":#(Load the available equipment from a file)

Step 5: If someone runs this program, call the "main" function. #(Keep showing a menu with options Display equipment, Rent equipment, Return equipment and Exit the program.)

Step 6: If User Chooses, "Display Equipment" option then call "display_available_equipment" Function.

Step 7: If User Chooses, "Rent Equipment" option then call "rent_equipment" Function.

Step 8: If User Chooses, "Return Equipment" option then call "return_equipment" Function.

Step 9: If User Chooses, "Exit Program" option then Loop should be terminated, and program should end.

Step 10: If any error occurs and any desired function is not called then, "An error" message is displayed and back to Step:5(Loop).

Step 11: If No error occurs, Desired output of called function is displayed for further Operations and Loop ends.

Step 12: End the Program.

2.2 Pseudocodes:

➤ Main.py Module:

Set equipment_data = file_operations.read_equipment_data("equipment.txt")

DO

Create Loop until user chooses to exit:

Display menu options:

1. Display available equipment
2. Rent equipment
3. Return equipment
4. Exit

Get user's choice

If choice equals 1:

Call display_functions.display_available_equipment(equipment_data)

Else if choice equals 2:

Call rental_functions.rent_equipment(equipment_data)

Call file_operations.write_equipment_data(equipment_data,
"equipment.txt")

Else if choice equals 3:

Call return_functions.return_equipment(equipment_data)

Call file_operations.write_equipment_data(equipment_data, "equipment.txt")

Else if choice equals 4:

Display exit message

Break the loop


```
Else:
    Display invalid choice message
End If
End loop

Start of the Program:

If this script is run as the main program:

    End start

END DO
```

➤ **rental_functions.py Module:**

Import necessary modules

Include display functions module.

Include file operations module.

DO

Function to create and save invoice (create_and_save_invoice):

Set invoice as an empty text.

For each item in items:

Extract equipment details (name, brand), quantity, and rental amount.

Append equipment details, quantity, and rental amount to the invoice.

End For

Calculate the total amount.

Generate invoice filename using the customer's name.

Open the invoice filename for writing.

Write the invoice content to the file.

Close the file.

Display "Invoice saved for 'invoice_filename'."

End Function

Function to process renting (process_renting):

Attempt:

Input customer's name and remove leading/trailing spaces.

Set total amount to 0.

Create an empty list for rented items.

Repeat indefinitely:

Display available items for rent.

Input the index of the item to rent.

If the index is within valid range:

Extract the chosen equipment.

Input rental quantity.

If the quantity is available and valid:

Update equipment quantity and calculate rental amount.

Add rented item details to the list.

Display success message.

Else:

Display error message for unavailable or invalid quantity.

Else:

Display error message for invalid item number.

Ask if the customer wants to rent another item.

If not, exit the loop.

If the rented items list is not empty:

Calculate the total amount.

Call the create_and_save_invoice function.

Update the equipment data file.

Exception for ValueError:

Display error message for invalid input.

End If

End Function

Start of the Program:

If this script is run as the main program:

Call the process_renting function.

End Start

End DO.

➤ **return_functions.py Module:**

Import necessary modules

Include display functions module.

DO

Function to generate and save return invoice

(generate_and_save_return_invoice):

Create return invoice content using equipment details, return quantity, return days, fine amount, and return datetime.

Input customer name and generate return invoice filename.

Open the return invoice filename for writing.

Write the return invoice content to the file.

Close the file.

Display "Return Invoice saved as 'return_invoice_filename'."

End Function

Function to return equipment (return_equipment):

Try:

Repeat indefinitely:

Display available equipment.

Input the index of the equipment to return.

If the index is within valid range:

Extract the chosen equipment.

Input returned quantity.

If returned quantity is positive:

Update equipment quantity.

Input return days and calculate fine if applicable.

Set fixed fine amount per day and maximum allowed rental days without fine.

If return days exceed maximum allowed days:

Calculate fine amount based on extra days.

```

        Else:
            Set fine amount to 0.
            Get current date and time.
            Call generate_and_save_return_invoice function.
            Display success message.
        Else:
            Display error message for invalid return quantity.
    Else:
        Display error message for invalid equipment number.
        Ask if the customer wants to return another item.
        If not, exit the loop.
    Except for ValueError:
        Display error message for invalid input.
End Function

Start of the Program:
    Call return_equipment function.
End Start
End DO.

```

➤ **display_functions.py Module:**

```

DO
    Function to display available equipment (display_available_equipment):
        Display formatted table header for available equipment.
        For each equipment in equipment_data:
            Display equipment number, name, brand, price, and quantity.
        End For
        Display table footer.
    End Function
End DO

```

➤ **file_operations.py Module:**

Import necessary modules

Include os module.

DO

Function to write equipment data to the file (write_equipment_data):

Open the specified file for writing.

For each equipment in equipment_data:

Write equipment's name, brand, price, and quantity to the file.

End For

Close the file.

End Function

Function to read equipment data from the file (read_equipment_data):

Create an empty list called equipment_data.

If the specified file exists:

Open the file for reading.

For each line in the file:

Split the line into name, brand, price, and quantity.

Append a dictionary containing equipment details to equipment_datalist.

Close the file.

Return equipment_data.

End Function

End DO.

2.3 Flowchart:

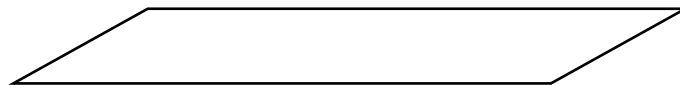
A flowchart is a diagram that shows the order of logical stages in a program. Simple geometric forms are used in flowcharts to represent processes, while arrows are used to demonstrate linkages and the flow of data.

The common symbols used in drawing flowcharts are as follows.

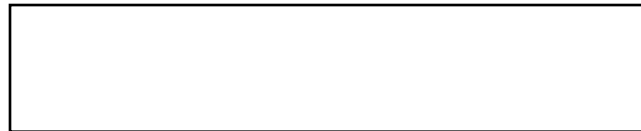
- a) **Start/End** : This symbol is used to indicate the start and finish of the algorithm at the beginning and end of each program.



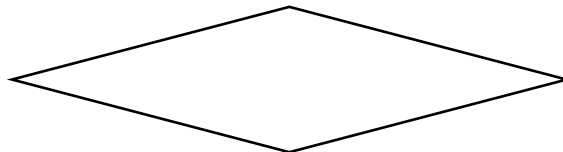
- b) **Input/ Output**: This Symbol is used while denoting program inputs and outputs.



- c) **Processing**: This symbol Indicates processes like mathematical operations and other processing operations.



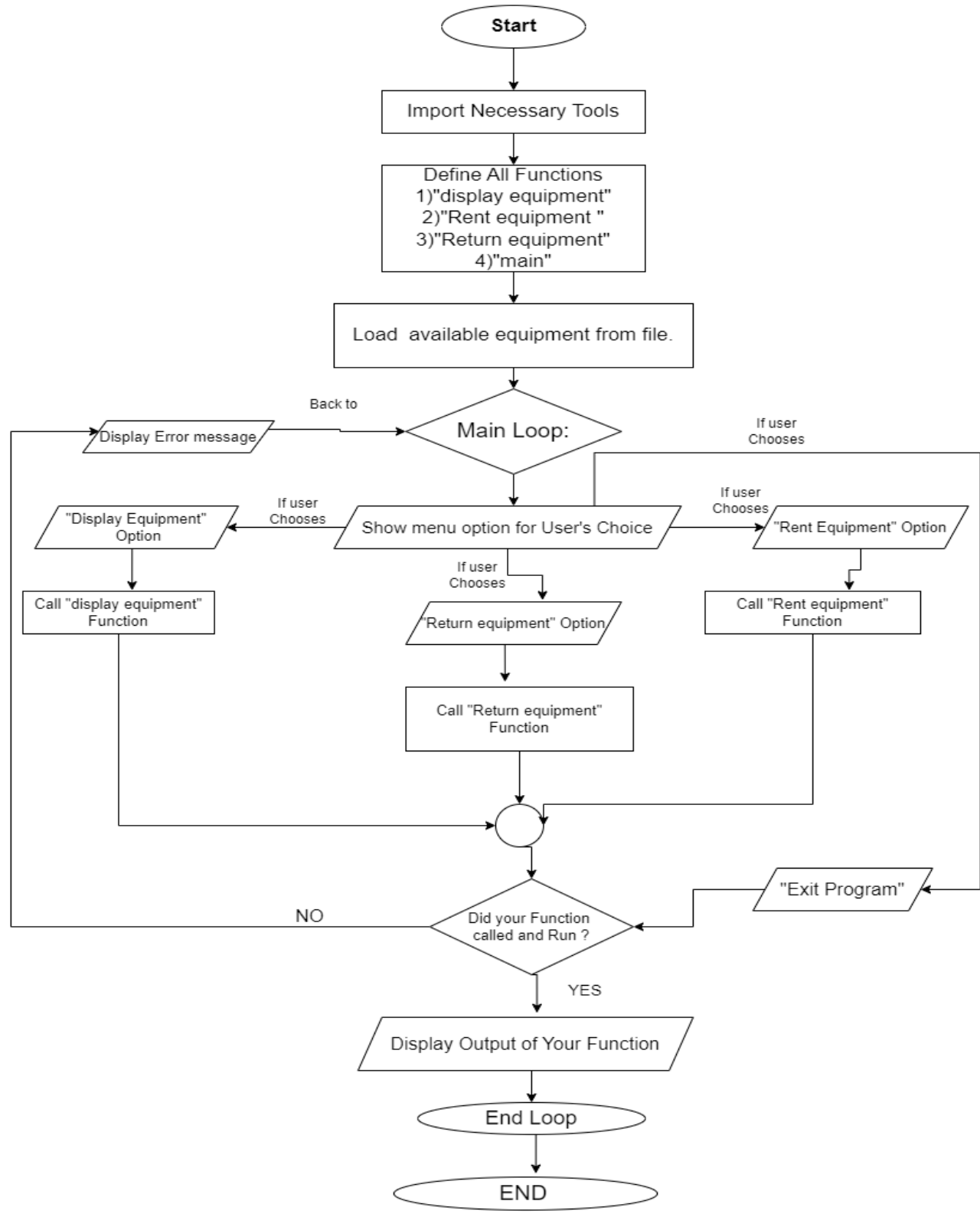
- d) **Decision**: This symbol stands for a program's decision statements, the responses to which are often Yes or No.



- e) **Arrow**: This symbol shows connections between various shapes (Tutorialspoint, 2023).



Flow chart of Program to Manage Equipment Rental Shop:



2.4 Data Structures:

- List: Data about the equipment that is available is kept in the `equipment_data` list. Each piece of equipment is displayed as a dictionary with entries for "name," "brand," "price," and "quantity."
- List of Dictionaries: Information about things that a client has rented is kept in the `rented_items` list. Every rented item is represented as a dictionary that contains details on the equipment, the number of days rented, and the rental cost.

Explanation:

- `equipment_data` List: This list contains data about the equipment that is readily available. The list consists of a glossary for each piece of equipment. The dictionary has four keys: "name," "brand," "price," and "quantity," which correlate to the name, brand, price, and quantity of the equipment.
- `rented_items` List of Dictionaries: Information about the goods a consumer has rented during a transaction is kept on this list. Each rented object is represented as a dictionary including data on the rented item's specifications, the number of rentals, and the rental cost.

3. Testing of Program.

3.1 Test-1: Show the implementation of Try, Except.

<u>Objective:</u>	To show the implementation of try, except on user's invalid inputs.
<u>Action:</u>	Step 1: Run the Program. Step 2: Enter the Invalid option of Menu. i.e., Providing non-existing option.
<u>Expected Result:</u>	When user chooses invalid input, the code which is inside except block and print an error message.
<u>Actual Result:</u>	Error message "Invalid choice. Please select a valid option."
<u>Conclusion:</u>	Testing Successful.

```
*IDLE Shell 3.10.9*
File Edit Shell Debug Options Window Help
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Asus\OneDrive\Desktop\FOC Final Program\Main.py =====
*****Bibek's Rental Shop*****
1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: 5
Invalid choice. Please select a valid option.
*****Bibek's Rental Shop*****
1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: B
Invalid choice. Please select a valid option.
*****Bibek's Rental Shop*****
1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: |
```

Fig: Test-1

3.2 Test-2: Selection rent and return of items.

<u>Objective:</u>	To Check the negative and non-existing value as input while renting or returning.
<u>Action:</u>	Step 1: Choose Rent Step 2: Enter Customer Name. Step 3: Enter negative value for Item Id. Step 4: Enter Non-existing value for Item ID (Same steps For Returning equipment.)
<u>Expected Result:</u>	When user chooses invalid input, an error message should be displayed.
<u>Actual Result:</u>	Error message "Invalid equipment number."
<u>Conclusion:</u>	Testing Successful.

```

*IDLE Shell 3.10.9*
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\Asus\OneDrive\Desktop\FOC Final Program\Main.py =====
*****Bibek's Rental Shop*****
1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: 2
Enter customer name: Bibek Thakur
*****
          Available Equipment
*****
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 90
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 50
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 500
7 | Architectural floodlight | Philips | $60.00 | 160
-----
Enter the equipment number to rent: -5
Invalid equipment number.
Do you want to rent another item? (yes/no): yes
*****
          Available Equipment
*****
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 90
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 50
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 500
7 | Architectural floodlight | Philips | $60.00 | 160
-----
Enter the equipment number to rent: 10
Invalid equipment number.
Do you want to rent another item? (yes/no): |

```

Fig: Test-2(For renting)

```
>>> ===== RESTART: C:\Users\Asus\OneDrive\Desktop\FOC Final Program\Main.py =====
=====Bibek's Rental Shop=====

1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: 3
=====
                        Available Equipment
=====
No. | Equipment Name          | Brand          | Price ($) | Quantity
-----
1   | Velvet Table Cloth      | Saathi         | $8.00     | 90
2   | Microphone Set          | Audio Technica | $129.00   | 70
3   | Disco Light Set         | Sonoff         | $122.00   | 60
4   | 7.1 Surround Sound Speaker Set | Dolby         | $299.00   | 50
5   | Dinner Table 8x5        | Panda          | $340.00   | 100
6   | Steel Chairs            | Steelcase      | $9.00     | 500
7   | Architectural floodlight | Philips        | $60.00    | 160
-----

Enter the equipment number to return: -7
Invalid equipment number.
Do you want to return another item? (yes/no): yes
=====
                        Available Equipment
=====
No. | Equipment Name          | Brand          | Price ($) | Quantity
-----
1   | Velvet Table Cloth      | Saathi         | $8.00     | 90
2   | Microphone Set          | Audio Technica | $129.00   | 70
3   | Disco Light Set         | Sonoff         | $122.00   | 60
4   | 7.1 Surround Sound Speaker Set | Dolby         | $299.00   | 50
5   | Dinner Table 8x5        | Panda          | $340.00   | 100
6   | Steel Chairs            | Steelcase      | $9.00     | 500
7   | Architectural floodlight | Philips        | $60.00    | 160
-----

Enter the equipment number to return: 15
Invalid equipment number.
Do you want to return another item? (yes/no):
```

Fig: Test-2(For returning).

3.3 Test-3: File generation of renting of items(s) (Renting multiple items(s):

<u>Objective:</u>	To Show complete renting process and show the rented items details in a text file
<u>Action:</u>	Step 1: Choose Rent Step 2: Enter Customer Name. Step 3: Enter rental equipment Number. Step 4: Enter rental quantity. Step 5: Show .txt file.
<u>Expected Result:</u>	An Invoice should be generated holding renting details and amount.
<u>Actual Result:</u>	Rent Invoice is saved on customer's name in .txt format with all details.
<u>Conclusion:</u>	Testing Successful.

IDLE Shell 3.10.9

File Edit Shell Debug Options Window Help

```

Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Asus\OneDrive\Desktop\FOC Final Program\Main.py =====
*****Bibek's Rental Shop*****

1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: 2
Enter customer name: Bibek Thakur
*****

Available Equipment
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 90
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 50
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 500
7 | Architectural floodlight | Philips | $60.00 | 160

Enter the equipment number to rent: 1
Enter rental quantity for Velvet Table Cloth: 30
Successfully rented 30 units of Velvet Table Cloth to Bibek Thakur.
Do you want to rent another item? (yes/no): yes
*****

Available Equipment
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 60
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 50
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 500
7 | Architectural floodlight | Philips | $60.00 | 160

Enter the equipment number to rent: 6
Enter rental quantity for Steel Chairs: 150
Successfully rented 150 units of Steel Chairs to Bibek Thakur.
Do you want to rent another item? (yes/no): no
Total Rental Amount for Bibek Thakur: $1590.00
Rent Invoice saved as 'Bibek Thakur rent invoice.txt'
*****Bibek's Rental Shop*****

```

IDLE Shell 3.10.9

File Edit Shell Debug Options Window Help

```

*****Bibek's Rental Shop*****

Available Equipment
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 90
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 50
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 500
7 | Architectural floodlight | Philips | $60.00 | 160

Enter the equipment number to rent: 1
Enter rental quantity for Velvet Table Cloth: 30
Successfully rented 30 units of Velvet Table Cloth to Bibek Thakur.
Do you want to rent another item? (yes/no): yes
*****

Available Equipment
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 60
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 50
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 500
7 | Architectural floodlight | Philips | $60.00 | 160

Enter the equipment number to rent: 6
Enter rental quantity for Steel Chairs: 150
Successfully rented 150 units of Steel Chairs to Bibek Thakur.
Do you want to rent another item? (yes/no): no
Total Rental Amount for Bibek Thakur: $1590.00
Rent Invoice saved as 'Bibek Thakur rent invoice.txt'
*****Bibek's Rental Shop*****

```

Fig: Test-3 Complete Renting process.

Bibek Thakur_rent_invoice - Notepad

File Edit Format View Help

```
===== Rent Invoice for Bibek Thakur =====  
Equipment: Velvet Table Cloth (Saathi)  
Quantity: 30  
Rental Amount: $240.00  
  
Equipment: Steel Chairs (Steelcase)  
Quantity: 150  
Rental Amount: $1350.00  
  
Total Rental Amount: $1590.00
```

Fig: Test-3 >>showing the rented items details in a text file

3.4 Test-4: File generation of return process of items(s)

<u>Objective:</u>	To Show complete return process and show the returned items details in a text file
<u>Action:</u>	Step 1: Choose Return Step 2: Enter Customer Name. Step 3: Enter return equipment Number. Step 4: Enter return quantity. Step 5: Show .txt file.
<u>Expected Result:</u>	An Invoice should be generated holding returning details along with late fine based on days holding more than 5.
<u>Actual Result:</u>	Return Invoice is saved on customer's name in .txt format with all details.
<u>Conclusion:</u>	Testing Successful.

```

*IDLE Shell 3.10.9*
File Edit Shell Debug Options Window Help
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Asus\OneDrive\Desktop\FOC Final Program\Main.py =====
*****Bibek's Rental Shop*****
1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: 3
*****
Available Equipment
*****
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 60
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 150
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 350
7 | Architectural floodlight | Philips | $60.00 | 160
-----
Enter the equipment number to return: 1
Enter returned quantity for Velvet Table Cloth: 40
Enter the number of days the equipment was rented for: 7
Enter customer name: Sima Singh
Return Invoice saved as 'Sima Singh_return_invoice.txt'
Successfully returned 40 units of Velvet Table Cloth.
Do you want to return another item? (yes/no): no
*****Bibek's Rental Shop*****
1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice:

```

Fig: Test-4 Complete Returning process.

Sima Singh_return_invoice - Notepad

```

File Edit Format View Help
===== Return Invoice =====
Equipment: Velvet Table Cloth (Saathi)
Returned Quantity: 40
Return Date: 2023-08-24 20:54:31
Number of Days Rented: 7
Fine Amount: $20.00

```

Fig: Test-4 >>showing the returned items along with late fine details in a text file.

3.5 Test-5: Show the update in stock of items(s)

<u>Objective:</u>	To Show 1) the quantity being deducted while renting the items (Update should be reflected in a .txt file as well.) 2) Show the quantity being added while returning the items (Update should be reflected in a .txt file as well)
<u>Action:</u>	Step 1: Choose Rent Step 2: Enter Customer Name. Step 3: Enter rent equipment Number. Step 4: Enter rent quantity. Step 5: Show .txt file. (Similar steps for returning)
<u>Expected Result:</u>	An Invoice should be generated holding renting details while renting or returning.
<u>Actual Result:</u>	Rent Invoice is saved on customer's name in .txt format with all details. Return Invoice is saved on customer's name in .txt format with all details.(For Return).
<u>Conclusion:</u>	Testing Successful.

During Renting.(Quantity being Reduced)

```

Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Asus\OneDrive\Desktop\FOC Final Program\Main.py =====
*****Bibek's Rental Shop*****


1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: 2
Enter customer name: Mukul Gupta
*****
Available Equipment
*****
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 100
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 150
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 350
7 | Architectural floodlight | Philips | $60.00 | 160

Enter the equipment number to rent: 1
Enter rental quantity for Velvet Table Cloth: 50
Successfully rented 50 units of Velvet Table Cloth to Mukul Gupta.
Do you want to rent another item? (yes/no): no
Total Rental Amount for Mukul Gupta: $400.00
Rent Invoice saved as 'Mukul Gupta_rent_invoice.txt'
*****Bibek's Rental Shop*****

1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: 1
*****
Available Equipment
*****
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 50
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 150
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 350
7 | Architectural floodlight | Philips | $60.00 | 160

```


Fig: Test-5>> The quantity being deducted while renting

 equipment - Notepad

File Edit Format View Help

```
Velvet Table Cloth, Saathi, 8.0, 100
Microphone Set, Audio Technica, 129.0, 70
Disco Light Set, Sonoff, 122.0, 60
7.1 Surround Sound Speaker Set, Dolby, 299.0, 150
Dinner Table 8x5, Panda , 340.0, 100
Steel Chairs, Steelcase, 9.0, 350
Architectural floodlight, Philips, 60.0, 160
```

Fig: Test-5>> Before Renting.


 Mukul Gupta_rent_invoice - Notepad

File Edit Format View Help

```
===== Rent Invoice for Mukul Gupta =====
Equipment: Velvet Table Cloth (Saathi)
Quantity: 50
Rental Amount: $400.00
```

Total Rental Amount: \$400.00

Fig: Test-5>> Rented 50 quantity to Customer.

 equipment - Notepad

File Edit Format View Help

```
Velvet Table Cloth, Saathi, 8.0, 50
Microphone Set, Audio Technica, 129.0, 70
Disco Light Set, Sonoff, 122.0, 60
7.1 Surround Sound Speaker Set, Dolby, 299.0, 150
Dinner Table 8x5, Panda , 340.0, 100
Steel Chairs, Steelcase, 9.0, 350
Architectural floodlight, Philips, 60.0, 160
```

Fig: Test-5>> quantity to deducted after renting .

During Returning.(Quantity being increased)

IDLE Shell 3.10.9

File Edit Shell Debug Options Window Help

```
*****
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 50
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 60
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 150
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 350
7 | Architectural floodlight | Philips | $60.00 | 160
-----

Enter the equipment number to return: 3
Enter returned quantity for Disco Light Set: 40
Enter the number of days the equipment was rented for: 8
Enter customer name: Aashok
Return Invoice saved as 'Aashok_return_invoice.txt'
Successfully returned 40 units of Disco Light Set.
Do you want to return another item? (yes/no): no

=====
*****Bibek's Rental Shop*****
=====

1. Display Equipments
2. Rent Equipment
3. Return Equipment
4. Exit
Enter your choice: 1
*****
Available Equipment
*****
No. | Equipment Name | Brand | Price ($) | Quantity
-----
1 | Velvet Table Cloth | Saathi | $8.00 | 50
2 | Microphone Set | Audio Technica | $129.00 | 70
3 | Disco Light Set | Sonoff | $122.00 | 100
4 | 7.1 Surround Sound Speaker Set | Dolby | $299.00 | 150
5 | Dinner Table 8x5 | Panda | $340.00 | 100
6 | Steel Chairs | Steelcase | $9.00 | 350
7 | Architectural floodlight | Philips | $60.00 | 160
-----

=====
*****Bibek's Rental Shop*****
=====
```


Fig: Test-5>> The quantity being added while returning.

equipment - Notepad

File Edit Format View Help

```
Velvet Table Cloth, Saathi, 8.0, 50
Microphone Set, Audio Technica, 129.0, 70
Disco Light Set, Sonoff, 122.0, 60
7.1 Surround Sound Speaker Set, Dolby, 299.0, 150
Dinner Table 8x5, Panda , 340.0, 100
Steel Chairs, Steelcase, 9.0, 350
Architectural floodlight, Philips, 60.0, 160
```


Fig: Test-5>> Before Returning.

 Aashok_return_invoice - Notepad

File Edit Format View Help

```
===== Return Invoice =====  
Equipment: Disco Light Set (Sonoff)  
Returned Quantity: 40  
Return Date: 2023-08-25 07:12:12  
Number of Days Rented: 8  
Fine Amount: $30.00
```

Fig: Test-5>> Return Invoice of Customer.(Including late fine)

 equipment - Notepad

File Edit Format View Help

```
Velvet Table Cloth, Saathi, 8.0, 50  
Microphone Set, Audio Technica, 129.0, 70  
Disco Light Set, Sonoff, 122.0, 100  
7.1 Surround Sound Speaker Set, Dolby, 299.0, 150  
Dinner Table 8x5, Panda , 340.0, 100  
Steel Chairs, Steelcase, 9.0, 350  
Architectural floodlight, Philips, 60.0, 160
```

Fig: Test-5>> After Returning the item, Quantity (Stock) being Increased.

4. Conclusion:

4.1 Summary of project:

The project involves developing an application for an Event Equipment Rental shop to manage its equipment inventory, rental transactions, and generate invoices. The program takes data about the equipment from a text file, It enables clients to rent the equipment, creates rental invoices, maintains the inventory of the equipment, manages returns of the equipment, determines penalties for late returns, and prepares return bills. The following are the main features:

- The program reads a text file containing equipment information and parses it to get the necessary information.(Reading and Parsing data)
- Customers can rent equipment, and the programme creates rental invoices with information on the equipment, the client, and the total amount due.(Renting process)
- After each rental transaction or Return transaction, the program adjusts the quantity.(Stock management).
- Customers can return equipment, and the program creates return invoices with information on the returned equipment, the customer, the date and time of the return, and any relevant fines for late returns.

4.2 Development Process:

- The overall Understandings of the project requirements was kept in mind, such as reading equipment data, controlling rentals, producing invoices, and updating inventory after each transaction in same way as stock decreases and increases in an actual shop.
- The overall Arrangement of the application's structural aspects, data storage systems, and interaction pattern was planned. And required data structures to store equipment information and transactional information were chosen.
- Execution: The code to read the equipment data from the text file was developed. The functions for equipment rental and return, calculations, invoice generation, displaying information and stock update (Increase/Decrease) were defined and executed along with main function to run the program.
- A simple to use interface was created so users can interact with the program. A menu bar which displays to choose to perform desired action such as viewing available equipment to rent, renting items, returning items, and Exit were developed as User Interface. Moreover, the system asks for customer name, equipment Id, etc in user friendly way.

- The program was tested at last thoroughly to find and address any faults or problems and the functionality of equipment rentals, return, stock changes, and invoice generation were tested and confirmed working.

4.3 Things Learned:

File handling: I was able to work with text files to read and interpret data, as well as writing data back to files after producing invoices or updating inventory.

- Data Structure: The overall Arrangement of the application's structural aspects, data storage systems, and interaction pattern was planned so it was easier to understand the concept of data structures after making this program.
- Transactional Logics: After working on this project, I was able to learn the real-world transactional logics and implemented them in this program. Similarly, I was able to manage rental and return procedures, determine fines, and handle various transactional phases.
- Error handling: I have learned the ability to deal with a variety of error situations, such as incorrect inputs, reading/writing problems in files, and different circumstances during transactions.
- Real-World Application: I was being aware of how operations can be performed in real world shop, and business challenges can be solved via software development.

4.4 Limitations and Challenges:

- Scalability: The current implementation uses a text file to store data. However, as the business grows, managing data in a text file becomes inefficient and prone to errors. Handling a larger inventory and customer base requires a more robust solution.
- Security: Storage of text files is not secure in terms of security. It's possible that private consumer data will be revealed. Additionally, there is no authentication system in place to manage application access.

- Data Validation and Integrity Checks: The text file-based technique is deficient in several areas. Since it is impossible to guarantee precise and reliable data entry, inconsistencies in inventory and transactions may result.

4.5 Solutions:

- Implementation of the database: Switch from utilising text files to a relational database system (like MySQL or PostgreSQL). Better data management, increased scalability, and support for data integrity and consistency are all provided by this.
- User Authentication: To provide safe access to the application, implement user authentication and authorisation. To do this, user accounts and roles with the proper permissions must be created.

5.References

References

Geeksforgeeks, 2023. *geeksforgeeks.org/introduction-to-microsoft-word*. [Online]
Available at: <https://www.geeksforgeeks.org/introduction-to-microsoft-word/>
[Accessed 23 August 2023].

geeksforgeeks, 2023. *https://www.geeksforgeeks.org/*. [Online]
Available at: <https://www.geeksforgeeks.org/what-is-an-algorithm-definition-types-complexity-examples/?ref=lbp>
[Accessed 20 August 2023].

Paraschiv, L., 2023. *fotc.com/blog/draw-io-online-guide*. [Online]
Available at: <https://fotc.com/blog/draw-io-online-guide>
[Accessed 24 August 2023].

Python Software Foundation, 2001-2023. *https://docs.python.org/3/library/idle.html#*. [Online]
Available at: <https://docs.python.org/3/library/idle.html#>
[Accessed 23 August 2023].

Tutorialspoint, 2023. *programming_methodologies*. [Online]
Available at:
https://www.tutorialspoint.com/programming_methodologies/programming_methodologies_flowchart_elements.htm
[Accessed 24 August 2023].

6. Appendix

6.1 Main.py Module

```
# Importing the necessary Modules.
```

```
import rental_functions
```

```
import return_functions
```

```
import display_functions
```

```
import file_operations
```

```
# The Main program loop
```

```
def main():
```

```
    equipment_data = file_operations.read_equipment_data("equipment.txt") # Loading
    equipment data from the file
```

```
    while True:
```

```
        print(
```

```
"=====
=====
=====")
```

```
        print(
```

```
            "*****Bibek's Rental
Shop*****")
```

```
        print(
```

```
"=====
=====
=====")
```

```
        print("1. Display Equipments")
```

```
        print("2. Rent Equipment")
```

```
        print("3. Return Equipment")
```

```
        print("4. Exit")
```

```
    # Getting user choice
```

```

choice = input("Enter your choice: ")
# Handling User's choice
if choice == "1": # For Displaying available Equipments in file.
    display_functions.display_available_equipment(equipment_data)

elif choice == "2": # For Rent and Update the equipment data in the file.
    rental_functions.rent_equipment(equipment_data)
    file_operations.write_equipment_data(equipment_data, "equipment.txt")

elif choice == "3": # For Return and Update the equipment data in the file.
    return_functions.return_equipment(equipment_data)
    file_operations.write_equipment_data(equipment_data, "equipment.txt")

elif choice == "4": # For Exiting the program

print("=====
=====")
    print("-----Thank You!-----")

print("=====
=====")
    break
else:
    print("Invalid choice. Please select a valid option.") # Handling invalid choices
# Entry point of the program
if __name__ == "__main__":
    main()

```

6.2 rental_functions.py Module:

```

# Importing the necessary Modules.

```



```

import display_functions
import file_operations

# Function to generate and save rent invoice
def generate_and_save_rent_invoice(customer_name, rented_items,
total_rental_amount):
    invoice = f"==== Rent Invoice for {customer_name} =====\n"
    # Adding rented items to the invoice
    for item in rented_items:
        equipment = item["equipment"]
        quantity = item["quantity"]
        rental_amount = item["rental_amount"]
        invoice += f"Equipment: {equipment['name']}
({equipment['brand']})\nQuantity: {quantity}\nRental Amount:
${rental_amount:.2f}\n\n"
    # Adding total rental amount to the invoice
    invoice += f"Total Rental Amount: ${total_rental_amount:.2f}\n"
    # For Generating invoice filename.
    invoice_filename = f"{customer_name}_rent_invoice.txt"

    with open(invoice_filename, "w") as file: # Used to Write the invoice to a file
        file.write(invoice)

    print(f"Rent Invoice saved as '{invoice_filename}'") # For confirmation
message
# Function to rent equipment
def rent_equipment(equipment_data):
    # Use of try and except to handle a ValueError exception.
    try:
        customer_name = input("Enter customer name: ").strip() # Getting
customer name
        total_rental_amount = 0 # Initialize total rental amount

```

```

rented_items = [] # To store rented items and quantities
# while Loop for selecting and renting equipment
while True:
    display_functions.display_available_equipment(equipment_data)    #
    Displaying available equipment
    # Get the index of the equipment to rent
    equipment_index = int(input("Enter the equipment number to rent: ")) -
1

    if 0 <= equipment_index < len(equipment_data):
        equipment = equipment_data[equipment_index]
        rental_quantity = int(input(f"Enter rental quantity for
{equipment['name']}: "))
        # for checking if quantity is available for rent or not and printing
        appropriate message.
        if equipment["quantity"] >= rental_quantity > 0:
            equipment_data[equipment_index]["quantity"] -= rental_quantity
            rental_amount = equipment["price"] * rental_quantity
            total_rental_amount += rental_amount # Update total rental
            amount
            rented_items.append({"equipment": equipment, "quantity":
            rental_quantity, "rental_amount": rental_amount})
            print(f"Successfully rented {rental_quantity} units of
            {equipment['name']} to {customer_name}.")
        else:
            print("Equipment is not available for rent, quantity is insufficient, or
            quantity is invalid. Try again.")
        else:
            print("Invalid equipment number.")

```

```

        another_rent = input("Do you want to rent another item? (yes/no):
    ").lower()
        if another_rent != "yes":
            break

    # Display and save rental invoice
    if rented_items:
        print(f"Total      Rental      Amount      for      {customer_name}:
    ${total_rental_amount:.2f}")
        generate_and_save_rent_invoice(customer_name,      rented_items,
    total_rental_amount)
        file_operations.write_equipment_data(equipment_data,
    "equipment.txt") # Update the file with new equipment quantities

    except ValueError: # if a ValueError is raised in the try block, except is
    executed.
        print("Invalid input. Please enter valid values.")

```

6.3 display_functions.py Module:

```

def display_available_equipment(equipment_data):
    """
        formatted table for Displaying the available equipment data.

        Parameters:
        equipment_data (list): This is a list of Dictionaries having equipment
        information.
            Each dictionary should have keys: 'name', 'brand', 'price',
            'quantity'.
    """
    # Making and Displaying Header for the equipment table.

    print("*****")
    print("                Available Equipment                ")

    print("*****")
    print("No. | Equipment Name          | Brand          | Price ($) | Quantity")
    print("-----")

```

```

# Iterate through equipment data and display each item
for i, equipment in enumerate(equipment_data, start=1):
    print(
        f"{i:<4} | {equipment['name']:<30} | {equipment['brand']:<15} | "
        f"${equipment['price']:<9.2f} | {equipment['quantity']:<8}")

    print("-----")

```

6.4 return_functions.py Module:

```

# Importing necessary modules
import datetime
import display_functions

# Function to generate and save return invoice
def generate_and_save_return_invoice(equipment, return_quantity, return_days,
fine_amount, return_datetime):

    # Create the return invoice content
    return_invoice = f"==== Return Invoice =====\nEquipment: {equipment['name']}"
    f"({equipment['brand']})\nReturned Quantity: {return_quantity}\nReturn Date:"
    f"{return_datetime.strftime('%Y-%m-%d %H:%M:%S')}\nNumber of Days Rented:"
    f"{return_days}\nFine Amount: ${fine_amount:.2f}"

    # Get customer name and generate return invoice filename
    customer_name = input("Enter customer name: ").strip()
    return_invoice_filename = f"{customer_name}_return_invoice.txt"

    with open(return_invoice_filename, "w") as file: # For Writing the return invoice to a
file
        file.write(return_invoice)

    print(f"Return Invoice saved as '{return_invoice_filename}'") # For confirmation
message

# Function to return equipment
def return_equipment(equipment_data):

```

```

# Use of try and except to handle a ValueError exception.
try:
    while True:
        display_functions.display_available_equipment(equipment_data)    # Display
available equipment
        equipment_index = int(input("Enter the equipment number to return: ")) - 1

        if 0 <= equipment_index < len(equipment_data):
            equipment = equipment_data[equipment_index]
            return_quantity = int(input(f"Enter returned quantity for {equipment['name']}: "))
            if return_quantity > 0:
                equipment_data[equipment_index]["quantity"] += return_quantity
                # Get return days and calculate fine if applicable
                return_days = int(input("Enter the number of days the equipment was rented
for: "))

                fine_per_day = 10 # Fixed Fine amount per day (adjustable as per need)
                max_allowed_days = 5 # Maximum allowed rental days without fine

                if return_days > max_allowed_days:
                    extra_days = return_days - max_allowed_days
                    fine_amount = fine_per_day * extra_days
                else:
                    fine_amount = 0

                current_datetime = datetime.datetime.now()

                generate_and_save_return_invoice(equipment, return_quantity,
return_days, fine_amount, current_datetime)

                print(f"Successfully returned {return_quantity} units of {equipment['name']}")

```

```

else:
    print("Invalid return quantity. Please enter a valid positive value.")

else:
    print("Invalid equipment number.")

another_return = input("Do you want to return another item? (yes/no): ").lower()
if another_return != "yes":
    break

except ValueError: # if a ValueError is raised in the try block, except is executed.
    print("Invalid input. Please enter valid values.")

```

6.5 file_operations.py Module:

```

import os

# Function to write equipment data to the file
def write_equipment_data(equipment_data, filename):
    with open(filename, "w") as file:
        for equipment in equipment_data:
            file.write(f"{equipment['name']}, {equipment['brand']}, {equipment['price']}, {equipment['quantity']}\n")

# Function to read equipment data from the file
def read_equipment_data(filename):
    equipment_data = []
    if os.path.exists(filename):
        with open(filename, "r") as file:
            for line in file:
                name, brand, price, quantity = line.strip().split(", ")
                equipment_data.append({"name": name, "brand": brand, "price": float(price), "quantity": int(quantity)})
    return equipment_data

```

