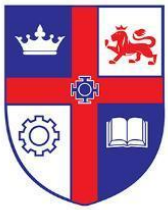
LONDON METROPOLITAN UNIVERSITY



islington college
(इस्लिङ्टन कलेज)

CS4051NI Fundamental of Computing

60% Individual Coursework - 2

2023-24 Spring

Student Name: Shamy Budhathoki

London Met ID: 22085618

College ID: NP01CP4S230083

Group: C20

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 marks of zero will be awarded.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

1.1) About the coursework

This coursework goal is to develop an application for an event equipment rental shop that rents out equipment to the customers for a cost fee which is charged on 5 day basis and fine is applied on a daily basis for late return. The shop provides various type of equipment to customers, so the application is created to keep track of the equipments. The main objective of the application is to read and display information from the text file .the application will show invoice during both renting return and process. The Stock will update after each transaction as when rented quantity decreases and when returned quantity increases. When a customer rents an item quantity changes from the shown quantity to decreased quantity and when the item is returned the quantity is increased. Unique file is create to separate the functionality and make the code easy to debug if some part of the code is needed and to make the code cleaner so that other programmer can understand the code easily. Files are separated in five parts main.py that handles main functionality of the user selected inputs if they want to rent or return the items or display the available equipments the shop has , readFile.py that reads the data stored in the text file, writeFile.py to write the user given data in the file and write it in transaction bill for rent and returning , operation.py that handles date-time and major updating and processing of the operation data, message.py shows information to the users or in structure information is displayed ,data.txt that stores information about the items brand price and stock quantity. Separate invoice is generated to separate between rent and return items. When renting invoice is generated that contains contact information of the customer and information of the item the customer is renting (item name, brand, date and time of rental, total amount to be paid). When renturning invoice is generated that contains contact information of the customer and information of the item the customer is renturning (item name, brand, date and time of returning, fine amount if customer is fined). Errors are handled to prevent incorrect input from the user. It is easy to add new items, quantity in the system as the information is stored in the text file and can be updated. The application is designed simple and user friendly so that no customer feels difficulty when using the application. Customer transaction and information is protected and can only be accessed by authorized person.

1.2) Tools used

1.2.1) Python IDLE

IDLE(Integrated Development and Learning Environment) is a tool for python programming. There are many IDEs but Python IDLE is quite suitable for begineer and intermediate programmers. Python IDLE is used as an interactive interpreter. (javatpoint, 2023).Python IDLE is a free and open source software so anyone can download and use it which can be personal or company use. python IDLE is compatible with major operating system such as Windows, macOS, Linux. Python IDLE is run in a shell that helps users to run short code easily. IDLE highlights where errors, auto indents after each line. It shows color for different variable ,keywords and errors which makes easy for begineers. Overall IDLE is a powerful software that provides various feature and services that makes writing, running code easier and the interface is simple to use.

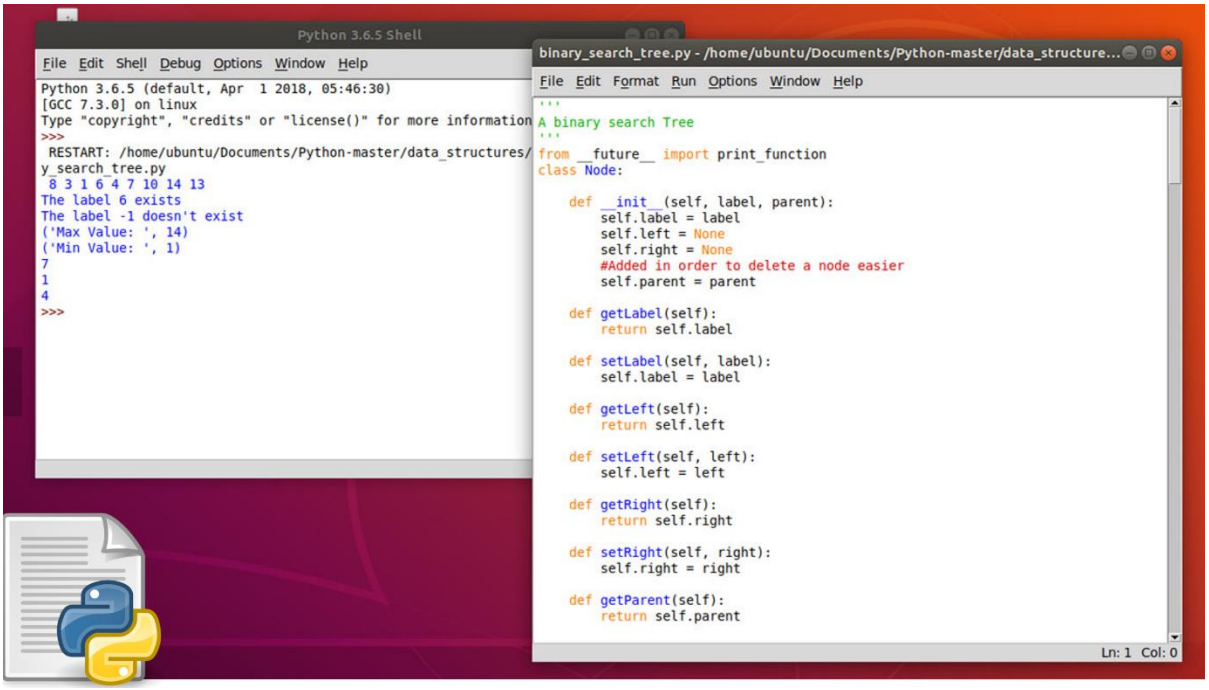


Figure 1: Python IDLE

1.2.2) MS-Word

Microsoft Word is an easy user friendly document creating software .It is a great tool for making documents for personal and company uses .word has many options for changing the looks and format of the document like fonts, color, size, pictures screenshot, chart,shapes. (BYJU'S, 2023) Word has word counter that counts total words written in the document and spelling checker that helps to correct the incorrect spellings. It highlights spelling errors written by the user with red color and spacing error with blue. User can make the document in any size like letter, A3, A4, A5. Microsoft Word allows user to save documents in different formats like PDF text,web pages .Giving refrence to the original author or source is simple compared to other documentation softwares .User can make the document in different language and there is a translate button that can translate single line, paragraphs or complete document to required language. Overall, Microsoft Word is an easy versatile and powerful tool for making documentation.

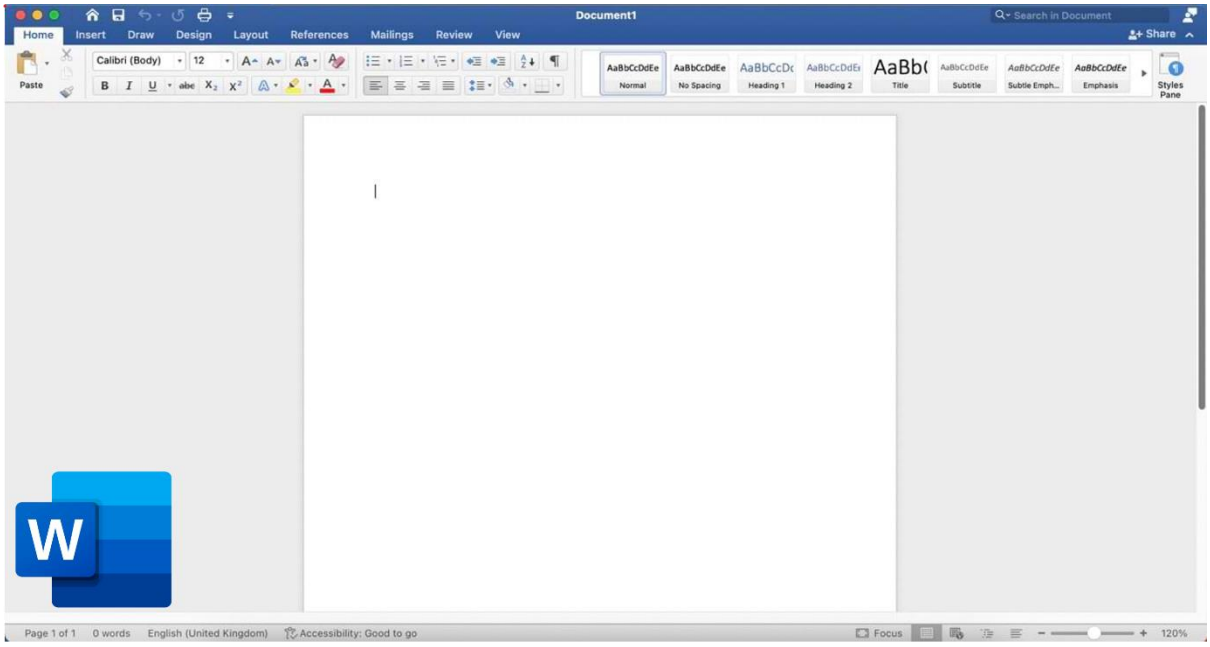


Figure 2: Ms-Word

1.2.3) Draw.io

Draw.io is a Free software for making diagrams,charts and flowcharts.Draw.io can be directly used from the website without any installation. It is available in all operating system as it can be used from websites.This software is used for various purposes for personal or Organization use. Without advance knowldege of creating diagram or figures draw.io is super easy to use (user-friendly). It is filled with various shapes ,box,arrow,lines,etc.Dragging and dropping an object or shape is simple and easy to use.Customizing the diagram with design and colors is also avaible. Sharing a diagram or project is simple as it can be shared through URL. The file can be Exported in many forms like PNG, AVG, PDF and many other forms.Diagram can be saved in many drives like google drive, oneDrive. Draw.io is an online web browser and application so it focus security and privacy. Multiple users can work on the same diagram while being on different devices.

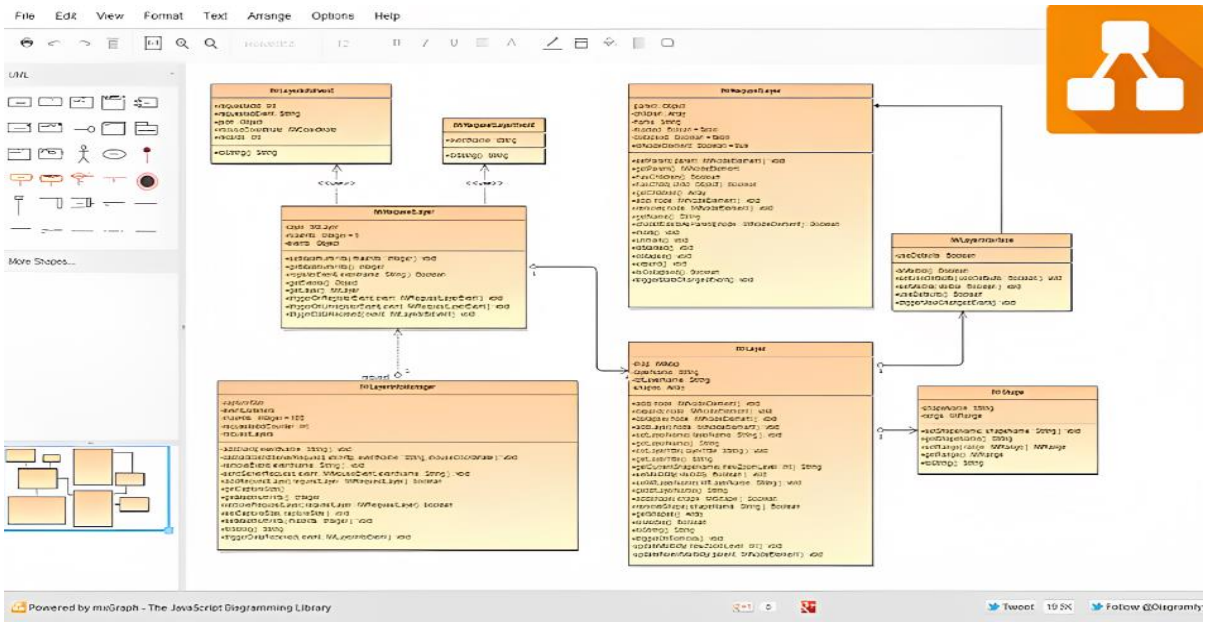


Figure 3:Draw.io

2. Algorithm

An algorithm is a set of instructions that must be followed to solve a problem. Algorithm is used to solve difficult problems in an easy understanding. Algorithm plays major role during development of the software as it makes the process quick and efficient. Algorithm should be in a well-defined structure that should help to take input and give out meaningful output. An algorithm should only have limited number of defined steps. When an algorithm is correct it will produce meaningful output.

Step 1: Read data from inventory text file

Step 2: Display welcome message and enter the loop to show main options

Step 3: Get the customer to choose option (1, 2, 3 or 4) which are rent, return display or exit

Step 4: When the customer selects option 1[Rent items]

Step 5: Get customer information and loop for rent, Display available items

Step 6: Get item and quantity customer needs, calculate total amount and generate rental invoice with transaction details

Step 7: Ask the customer if they want to continue renting, if customer don't want to rent update the Inventory

Step 8: When the customer selects option 2[Return items]

Step 9: Get customer information and loop for return, Display available items

Step 10: Get item and quantity customer needs, calculate fine amount if customer is late to return and generate return invoice with transaction details

Step 11: Ask the customer if they want to continue renturning, if customer don't want to return update the Inventory stock

Step 12: When the customer selects option 3[Display inventory], equipment inventory should be displayed

Step 13: When the customer selects option 4[Exit], exit message should appear and program ends

3. Flow Chart

A flowchart is a representation of a process or an algorithm in a symbolic way so that it helps to understand how a complex and hard problem is done visually. Flow chart is also made to show the process in a documentation way. A flowchart is picture of shapes that shows the flow of the program and it also helps to check if the flow of algorithm is correct or not .Different shapes are used to represent different function in the flowchart oval is used during start or end of the program, parallelogram is used to show input or output of the operation, Rectangle is used top show processing function of the program, Rhombus is used for decision making and branching ,Circle is used for connecting two parts of the program in a same page and finally arrow is used to show flow of the program.

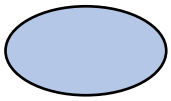
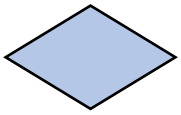
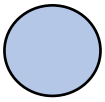
SHAPE	NAME	MEANING
	Oval	Start or end of the program.
	Parallelogram	Input or Output of operations.
	Rectangle	Processing function of a program.
	Rhombus	Decision making.
	Circle	Connecting two parts of program on same page.
	Arrow	Shows the flow of the code.

Table 1:Flowchart table

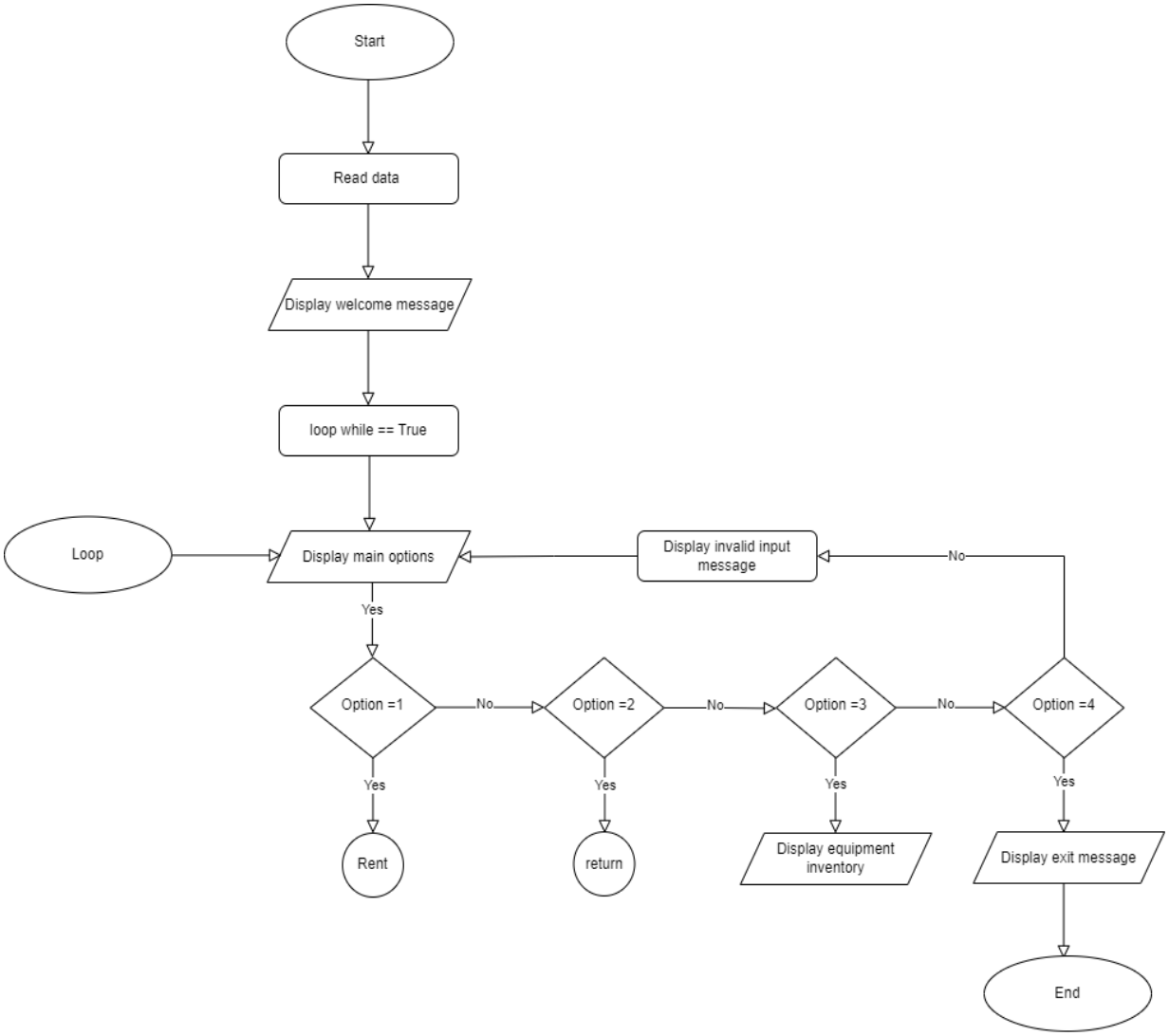


Figure 4:Main flowchart

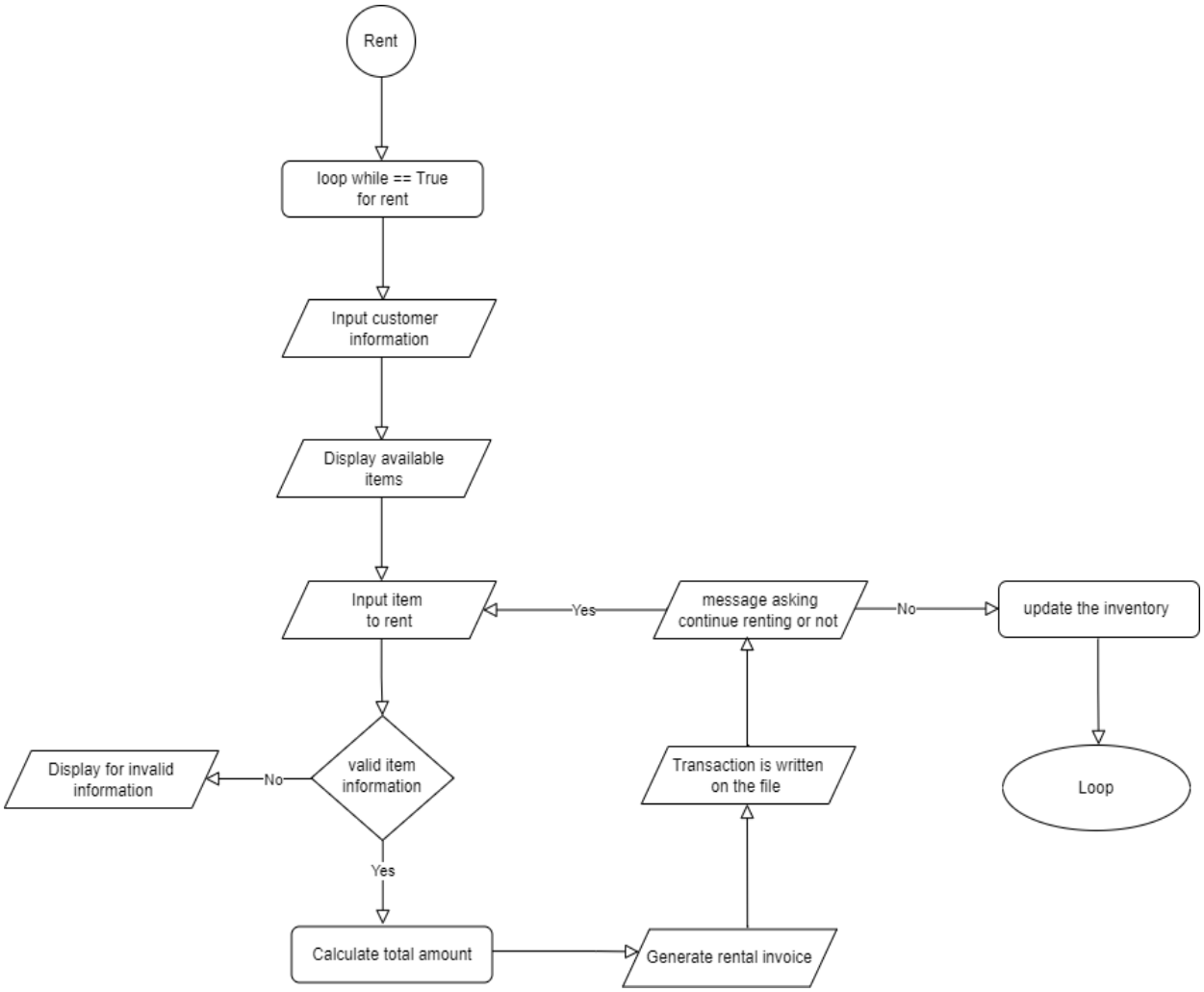


Figure 5: Rent flowchart

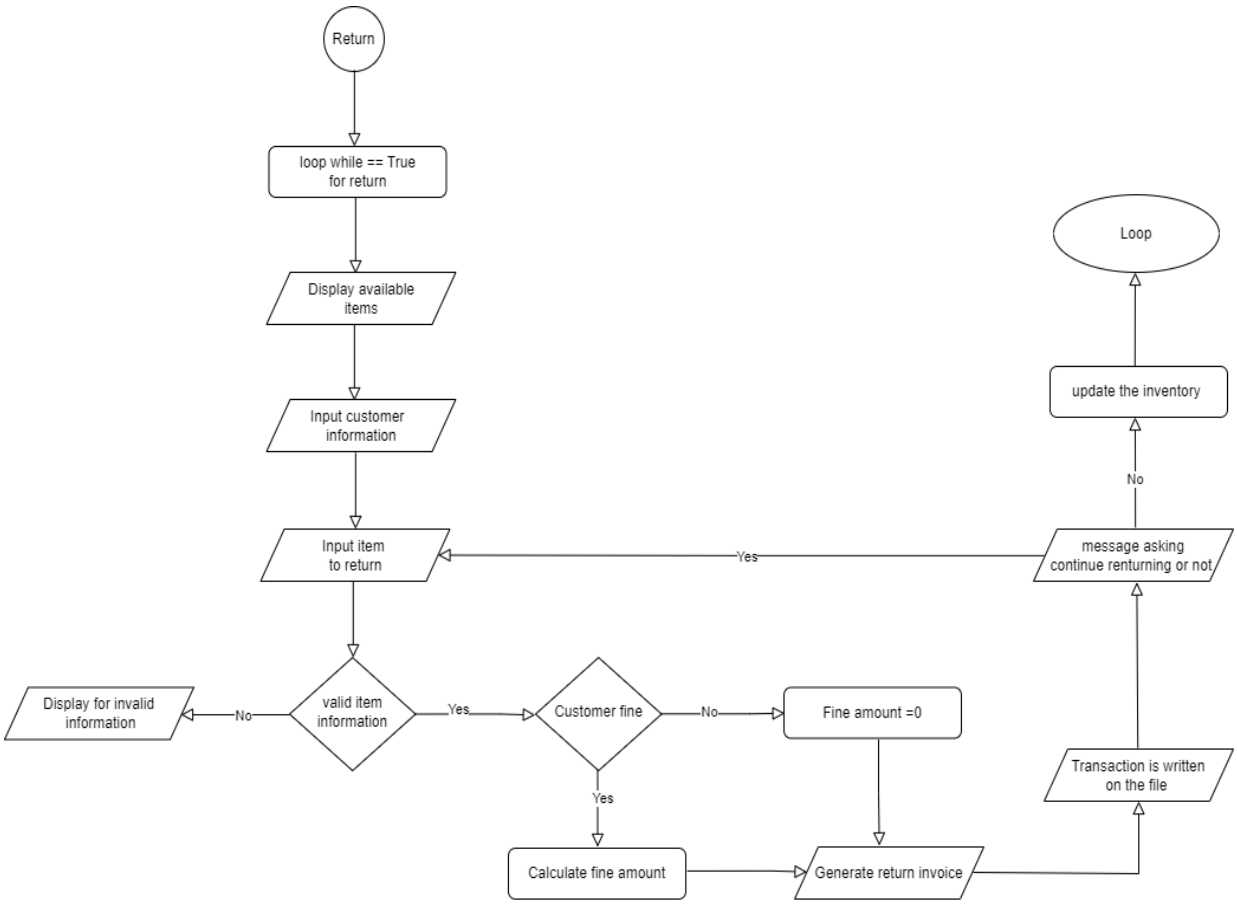


Figure 6:Rent flowchart

4. Pseudocode

4.1.1) Main.py

Import all files required

Create a main function

Call readData function to get data from the readfile

Call display_welcome_message function to displaying welcome message from message file

Initialize option to zero

While True

Call display_option_message from the message file

Try

Input option

Except ValueError

Display Invalid input!!! please enter a valid Number

If option is smaller or equals zero or greater than four **Then**

Display Invalid input. Please enter a value from one to four

If option equals one **Then**

Input customer name

Input address

try

Input phone number

Except ValueError

Display Invalid input!!! please enter a valid Number

Continue

Input email address

Call displayData from operation file

Initialize total_amount to zero

Initialize total_quantity to zero

Create a list name store

While True

Try

Input row or S.N to rent from

Except ValueError

Display Invalid input!!! please enter a valid Number

Continue

IF row is greater or equals to one and row is less and equals to length of Data **Then**

Try

Input quantity to rent

Except ValueError

Display Invalid input!!! please enter a valid Number

Continue

Set available_quantity to integer value of data at index row minus one and Index three

If quantity is greater or equal to one and is less than equals to available_quantity **Then**

Add row and quantity in store list

Set item to value of data at index row minus one

Assign action to

Calculate total_amount by multiplying quantity and item in index two

Add total_quantity sum with quantity

Set rent_invoice to generate_rental_invoice function with arguments customer_name, action, total_amount, quantity, store, data,phone_number,address,email_address in message file

Set data to update_Data function with arguments data, row, quantity in operation file

Call write_Transaction_rent function with arguments rent_invoice, customer_name From WriteFile file

Call write_Data function with arguments data from WriteFile file

End if

Else

Display Insufficient stock!!! Please enter a valid stock and show available quantity

End if

Else

Display Invalid S.N. Please enter a valid S.N from the table

Input ask user to continue renting or not

If lowercase value of ask is n or no **Then**

Display your invoice has been generated

break

End if

End While

Elif option equals two

Create returned_items list

Create returned_quantities list

Create fine_amounts list

Create rental_durations list

Input Customer name

Input address

Try

Input phone number

Except ValueError

Display Invalid input!!! please enter a valid Number

Continue

Input email address

Call displayData with argument data from Operation file

Set isTrue to True

While isTrue:

Try

Input row or S.N to return from

Except ValueError

Display Invalid input!!! please enter a valid Number

Continue

IF row is greater or equals to one and row is less and equals to length of Data **Then**

Try

Input returned quantity

Except ValueError

Display Invalid input!!! please enter a valid Number

Continue

Try

INPUT rental_duration

Except ValueError

Display Invalid input!!! please enter a valid Number

Continue

Set item to value of data at index row minus one

Add item and row in returned_items list

Add returned_quantity in returned_quantities list

Add rental_duration in rental_durations list

Set fine_per_day to integer of item in index two

Set rental_days to five

Calculate late_days to rental_duration subtract rental_days

Calculate fine_amount to late_days multiply fine_per_day

Add fine_amount in fine_amounts list

Set action to Return

Set return_invoice to enerate_return_invoice with arguments returned_items, customer_name, action, returned_quantities,

fine_amounts, phone_number, rental_durations,address,email_address in message file

Set data to update_Data with arguments data, row, minus returned_quantity

Call write_Transaction_return with arguments return_invoice, customer_name from the WriteFile file

Call write_Data with argument data the inventory from WriteFile file

End if

Else

Display Invalid S.N. Please enter a valid S.N from the table

Input ask user to continue renting or not

If lowercase value of ask is n or no **Then**

Display your invoice has been generated

break

End if

End Elif

End While

Elif option equals three

Call displayData with argument data from Operation file

End Elif

Elif option equals four

Call display_exit_message function from Message file

Break

End if

END WHILE

END main function

4.1.2) ReadFile.py

Create function readData

OPEN file Inventory.txt for reading

Create list name data

For Each line in readlines from the file

Set row to split line after , and then removing new line

Set row at index two to integer of row index two after removing dollar symbol

Add data to the row

End For

Close file

Return data

END function readData

4.1.3) WriteFile.py

Import Operation file

Create function write_Data with attribute data

OPEN file Inventory.txt for writing

for row in data

Create list changed_row

For Each indx in rangeof row length

Set item in row of indx

If indx equals two **Then**

Add dolar to sum string of item in changed_row list

End if

Else

Add string of item in changed_row list

End For

Write the output of joining changed_row with , with newline to the file

END For

Close file

END function write_Data

Create function write_Transaction_rent with arguments rent_invoice, customer_name

Get times to string file_time from Operation file

Set transaction_file_name to addition of customer_name, Rent, times, transaction.txt

OPEN file Inventory.txt for writing

Write for rent_invoice and with newline to the file

Close file

END function write_Transaction_rent

Create function write_Transaction_return with attributes return_invoice, customer_name

Get times to string file_time from Operation file

Set transaction_file_name to addition of customer_name, Return, time, transaction.txt

OPEN file Inventory.txt for writing

Write for renturn_invoice and with newline to the file

Close file

END function write_Transaction_return

4.1.4) Operation.py

Import datetime file

Create function displayData with attribute data

Display Equipment Inventory

Display table header

For Each item_num in range of the length of data

Set row in data at index item_num

Set S.N to item_num and add one

Display s.n,item,brand,price and quantity in a table

END function displayData

Create function update_Data with attributes data, row and quantity

Set data at row minus one and index three to string value of difference between integer of data at index row minus one and index three and quantity

Return data

END function update_Data

Create function function_time

Set year to string of current year from datetime

Set month to string of current month from datetime

Set day to string of current day from datetime

Set hour to string of current hour from datetime

Set minute to string of current minute from datetime

Set final_time to addition of year, month, day, hour, minute

Return final_time

End function function_time

Create function file_time

Set year to string of current year from datetime

Set month to string of current month from datetime

Set day to string of current day from datetime

Set hour to string of current hour from datetime

Set minute to string of current minute from datetime

Set file_time to sum string of year, month, day, hour, minute

Return file_time

End function file_time

4.1.5) Message.py

Import Operation file

Create function display_option_message

Display Event Equipment Rental shop

Display in docstring S.N, options and option should show Rent, return, display, exit options

End function display_option_message

Create function display_welcome_message

Display Welcome message

End function display_welcome_message

Create function display_exit_message

Display exit Thank you message

End display_welcome_message

Create function generate_rental_invoice with attributes customer_name, action, total_amount, quantity, store, data,phone_number,address,email_address

Set invoice to show Event Equipment Shop at top then, show in BILLING DETAILS, RENTAL INVOICE,Customer ,Rental date,Address ,Action,Phone Number,Email Address in required order

Add invoice to show heading for table which are S.N, Equipment, Brand, Price, Quantity

Initialize sn to one

For items in store

Set row to items in index zero

Set quantity to items in index one

Set item_info to data in row minus one

Add invoice to show item customer rented which are S.N, Equipment, Brand, Price, Quantity

Increment sn to one

End For

Add invoice to show Total Amount and out contact informations

Return invoice

End function generate_rental_invoice

Create function generate_return_invoice with attributes items, customer_name, action, returned_quantities, fine_amounts, phone_number, rental_durations,address,email_address

Set invoice to show Event Equipment Shop, BILLING DETAILS, RENTURN INVOICE,Customer, Return date, Address, Phone Number, Email Address

Add invoice to show S.N, Equipment, Brand, Returned Quantity, Rental Duration

Initialize total_fine as zero

For Each index in range of the length of items

Set item and row to items in index

Set returned_quantity to returned_quantitie in index

Set fine_amount to fine_amounts in index

Set total_fine sum of fine_amount and total_fine

Set rental_duration to rental_durations in index

Add invoice to show item customer rented which are S.N, Equipment, Brand, Price, Quantity

End For

If total_fine greater than zero **Then**

Add invoice to show Fine Amount and out contact informations

Else

Add invoice to show Fine Amount to zero dollar and our contact informations

Return invoice

End function generate_return_invoice

5. Data Structures

5.1) Implemented data structures

5.1.1) List example

List are majorly used in the code as it is used to store different data like quantity, item, returned_items and more.List can be changeable(mutable) and can hold different types of data. In my code data is used to store Equipment, brand, price and quantity.

```
def readData():
    file = open("Inventory.txt", "r") #opening inventory file
    data = [] #list to store data
```

Figure 7:List data structures

5.1.2) Tuple example

In the store list tuple is used to store rented items with the quantity.Tuple cannot be changeable(immutable) and can hold different data types.In my code in store list row and quantity is added to the list as a tuple

```
        total_amount = 0
        total_quantity = 0
        store = [] #creating list to store row and quantity
```

Figure 8: Store list

```
    if 1 <= quantity <= available_quantity:
        store.append((row, quantity))#adding it in store list
        item = data[row - 1]
```

Figure 9: Tuple example

5.2) Choice of the data structure

The data structures that I choose are List and tuples as for. List is used to store information about the items in inventory like equipment name brand price,etc. List allows to store collection of items in order. List is mainly used in the code as it is mutable and inventory data needs to be changed and updated frequently. List are also used to store returned quantities, fine amount and rental duration. Tuple allows to store collection of items in order that cannot be changed. Tuple is used to store rented items and quantities in the store list. Tuple is used in the code as only items or values that does not need to be changed are stored in it.

5.3) Other data structures with examples

Dictionary and Sets are other data structures which are not used in the code. Dictionary is used to store data values in key value pair. It is changeable(mutable) and stores collection of different data types.Example:

Bike = {

“Brand”: “Yamaha”,

“Model”: “MT-07”,

“Weight”: 163,

“Released year”: 2021}

Sets are used to store different items in a same variable with unique elements can only be stored.Example:

Food ={“pizza”,”burger”,”roti”}

5.4) Primitive and Collection data types

Primitive data types are the data types that are build in and provided by the programming language.Types of primitive data types are as follows

1.Integer: Whole numbers like ‘1’,’7’,’-3’.

```
    try:
        row = int(input("\n"+"1. Enter the S.N to Rent from : "))
```

Figure 10: Integer type

2.String: characters or word like ‘hi’,’rahul’

```
#Getting customer information
customer_name = input("Enter the name of the customer Returning : ")
address=input("Enter your address : ")
```

Figure 11: String type

3.Float: decimal number’0.01’,’233.99’

4.Boolean: true or false values,’True’,’False’

Collection data types allows to put multiple values together which may be unique,same ,different,changeable non changeable.Tpes of collection data types are as follows

1.List: to store collection of elements which are mutable’[Bryant,Kanye,Jones]

2.Tuples: to store collection of elements which are immutable’(1,695)’

3.Dictionaries: to store data values in key value pair ‘{‘item’:’brand’:900}

4.Sets: to store different items in a same variable with unique elements ‘{0,3}’

6. Program

Program on the basis of UI welcome message is displayed at first to the user Option selection menu is then shown where options are given in S.N 1,2,3 or 4 for renting returning, displaying inventory and exit.

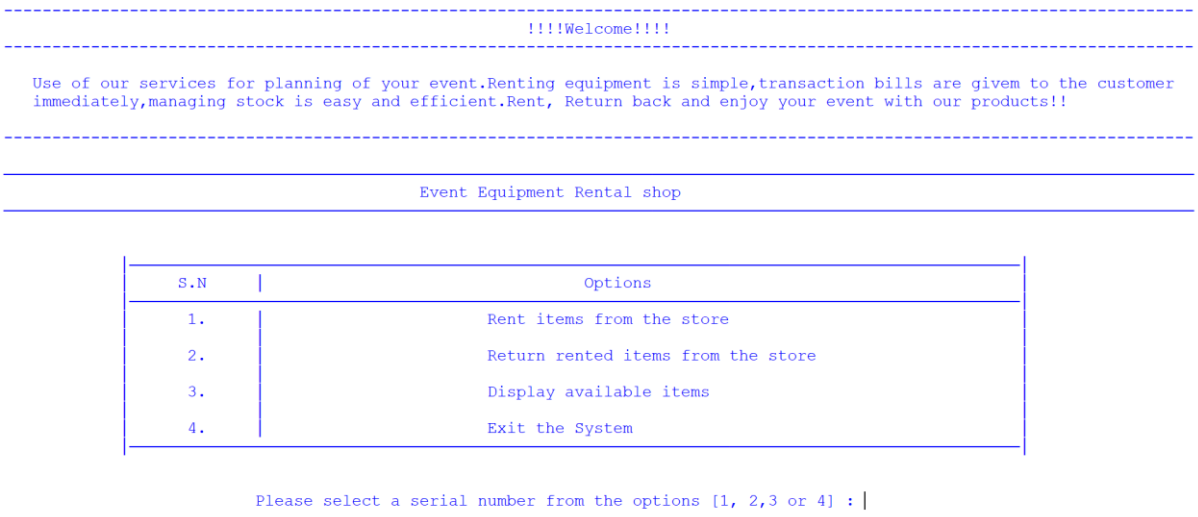


Figure 12: Welcome message

When the user selects option 1 then Available items for rent is displayed. User contact information is asked regarding their name number address phone number and email address. Available items is then displayed with S.N for input .The input is then checked if valid it checks if the item is in stock or not. Quantity to rent is then asked to the user. Total amount is then calculated and rental invoice is generated .The inventory is then updated and transaction detail is shown in txt file

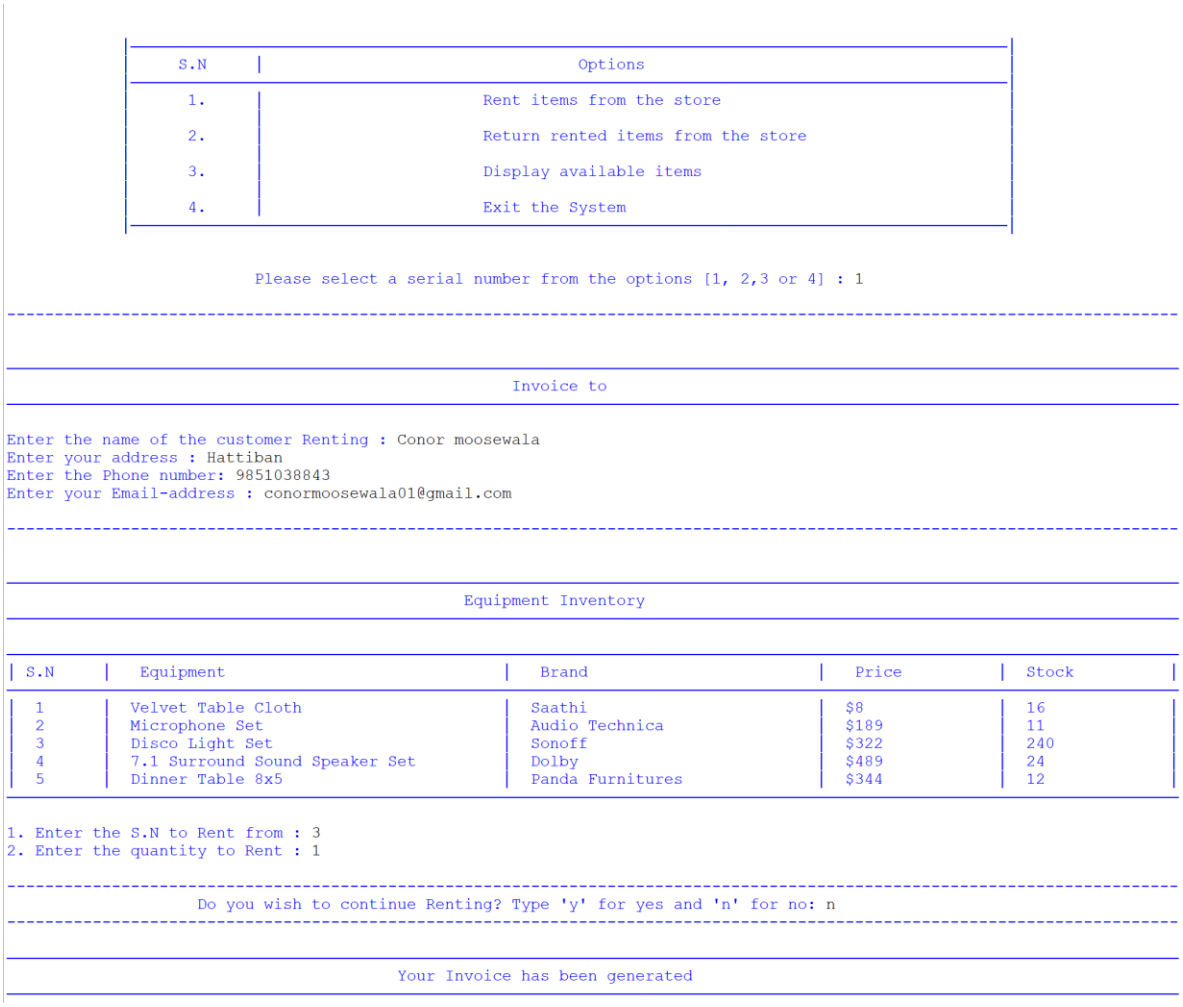


Figure 13:renting

When the user selects option 2 then Available items for return is displayed. User contact information is asked regarding their name number address phone number and email address. Information regarding to returning item is then shown and users gives input. Fine amount is then calculated if fined then fine amount is shown if not fined then fine amount is shown 0 and rental invoice is generated. The inventory is then updated and transaction detail is shown in txt file.

```
| S.N |                    Options                    |
|  1. |          Rent items from the store            |
|  2. |          Return rented items from the store   |
|  3. |          Display available items              |
|  4. |          Exit the System                      |

        Please select a serial number from the options [1, 2,3 or 4] : 2
-------------------------------------------------------------------------------------------------------------------------

                                        Invoice to

Enter the name of the customer Returning : Conor Moosewala
Enter your address : Hattiban
Enter the Phone number: 9851038843
Enter your Email-address : conormoosewala01@gmail.com
-------------------------------------------------------------------------------------------------------------------------

                                    Equipment Inventory

| S.N |  Equipment                    |  Brand              |  Price  | Stock |
|  1  | Velvet Table Cloth            | Saathi              | $8      | 16    |
|  2  | Microphone Set                | Audio Technica      | $189    | 11    |
|  3  | Disco Light Set               | Sonoff              | $322    | 239   |
|  4  | 7.1 Surround Sound Speaker Set| Dolby               | $489    | 24    |
|  5  | Dinner Table 8x5              | Panda Furnitures    | $344    | 12    |

1. Enter the S.N to Return from : 3
2. Enter the quantity to Return : 1
3. Please enter Your rental duration [in days]: 9
-------------------------------------------------------------------------------------------------------------------------
        Do you wish to continue Returning? Type 'y' for yes and 'n' for no: n

                                Your Invoice has been generated
```

Figure 14: Returning

After renting or returning a txt file is then generated which shows transaction invoice.The invoice is shown in the textfile with unique filenames and when Option 3 is selected updated inventory is shown

```
Event Equipment Rental shop

| S.N | Options |
| 1.  | Rent items from the store |
| 2.  | Return rented items from the store |
| 3.  | Display available items |
| 4.  | Exit the System |

Please select a serial number from the options [1, 2,3 or 4] : 3
---------------------------------------------------------------------------------

Equipment Inventory

| S.N | Equipment                    | Brand            | Price | Stock |
| 1   | Velvet Table Cloth           | Saathi           | $8    | 16    |
| 2   | Microphone Set               | Audio Technica   | $189  | 11    |
| 3   | Disco Light Set              | Sonoff           | $322  | 240   |
| 4   | 7.1 Surround Sound Speaker Set | Dolby          | $489  | 24    |
| 5   | Dinner Table 8x5             | Panda Furnitures | $344  | 12    |
```

Figure 15: Display updated items

Rent and return invoice are as follows

```
---------------------------------------------------------------------------------
                              Event Equipment Shop
---------------------------------------------------------------------------------
BILLING DETAILS :                                       RENTAL INVOICE :

Customer : Conor Moosewala                               Rental date: 2023-8-24/1:2
Address : Hattiban                                       Action : Rent
Phone Number : 9851038843
Email Address : conormoosewala01@gmail.com

---------------------------------------------------------------------------------
| S.N    |   Equipment             |   Brand          |   Price     |  Quantity   |
---------------------------------------------------------------------------------
| 1      | Disco Light Set         | Sonoff           | $322        | 1           |
---------------------------------------------------------------------------------

                                                         Total Amount : $322


QUESTIONS?

Email us at eventrentalshop@gmail.com or Contact Us at 01-994869

                            THANK YOU FOR YOUR BUSINESS

---------------------------------------------------------------------------------
```

Figure 16:Rent invoice

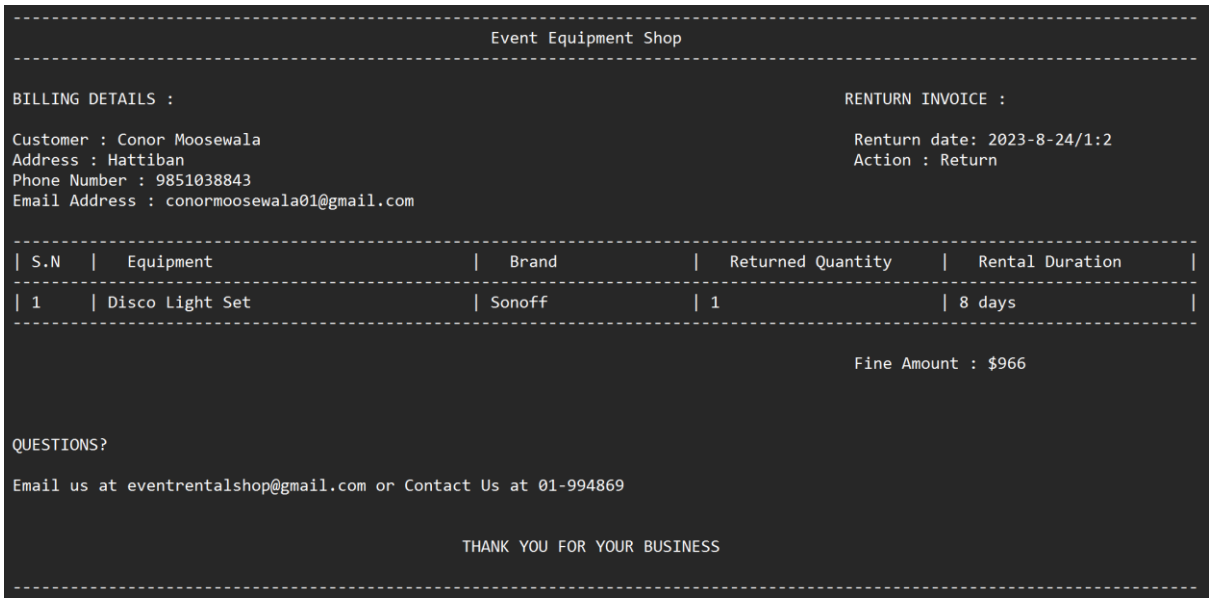


Figure 17:Return invoice

Program termination after each rent and return action User is asked if they want to continue returning or rent if yes then program continues for renting or returning if no is selected then the loop ends. If user selects Option 4 which is exit then the program ends with displaying exit message

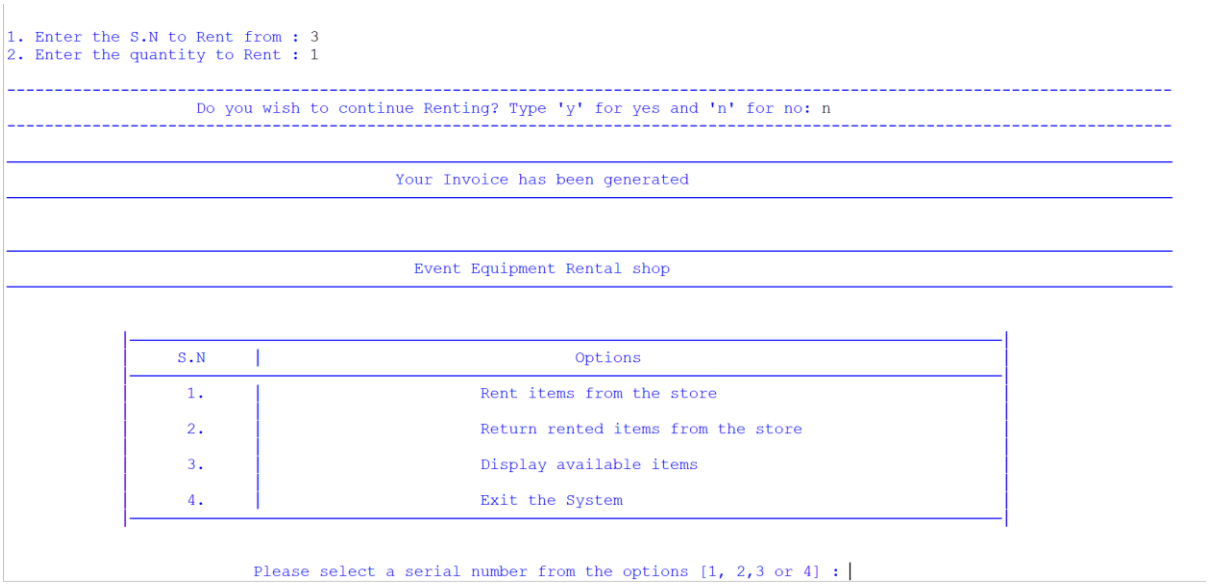


Figure 18:User selects no

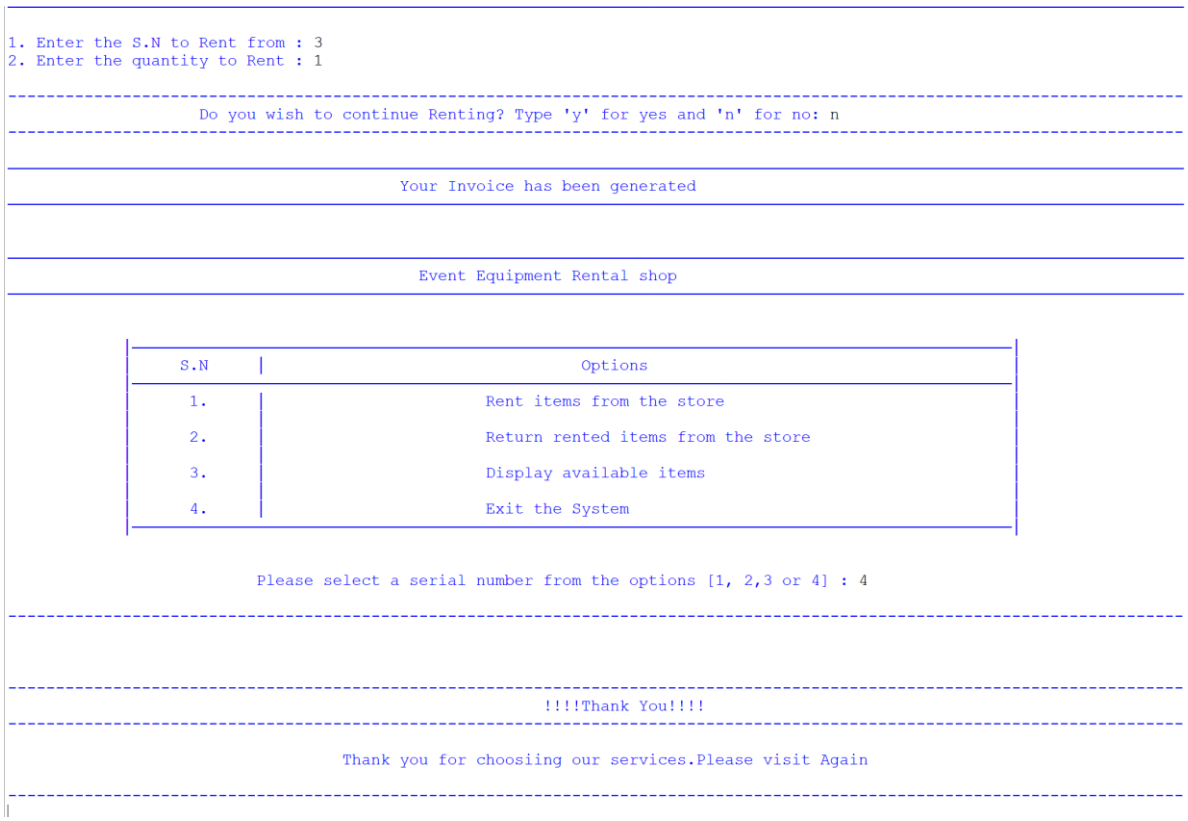


Figure 19:Exit

When user input invalid data program must check the input and should display error message like when string value is entered instead of integer the program handles the input properly without crashing and displays error message.

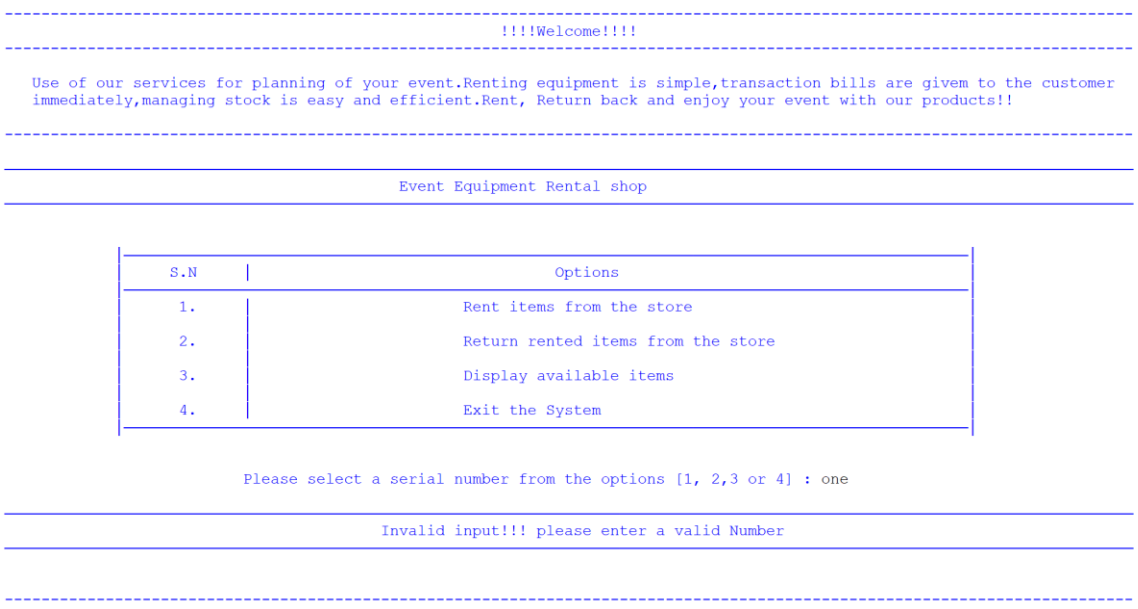


Figure 20:Invalid input

7. Testing

7.1) Test 1

Objective	To show implementation of try except, when invalid input is provided error message should be displayed
Action	try except is implemented, Invalid input is provided and error message is displayed after that
Expected Result	Invalid message should be displayed for both
Actual Result	Invalid message is displayed for both
Conclusion	Test was successful

Table 2:Test 1

When invalid data type is entered for example string is entered where the input needs to be in integer invalid message is displayed as try except for value error works

```
------------------------------------------------------------------------------------------------------------------------
                                Invalid input.Please enter a value from 1 to 4
------------------------------------------------------------------------------------------------------------------------

________________________________________________________________________________________________________________________
                                        Event Equipment Rental shop
________________________________________________________________________________________________________________________

|_____________________________________________________________________________|
|    S.N    |                              Options                            |
|___________|_____________________________________________________________________|
|    1.     |                    Rent items from the store                      |
|    2.     |                    Return rented items from the store             |
|    3.     |                    Display available items                        |
|    4.     |                    Exit the System                                |
|___________|_____________________________________________________________________|

                    Please select a serial number from the options [1, 2,3 or 4] : 1
------------------------------------------------------------------------------------------------------------------------

________________________________________________________________________________________________________________________
                                                Invoice to
________________________________________________________________________________________________________________________

Enter the name of the customer Renting : Izzy Kumar Shah
Enter your address : Lalitpur
Enter the Phone number: no phone number

________________________________________________________________________________________________________________________
                                Invalid input!!! please enter a valid Number
________________________________________________________________________________________________________________________
```

Figure 22: Use of try except

When invalid input is entered for example there are only 4 options in the option table when more than 4 or less than 1 is entered error message is displayed.

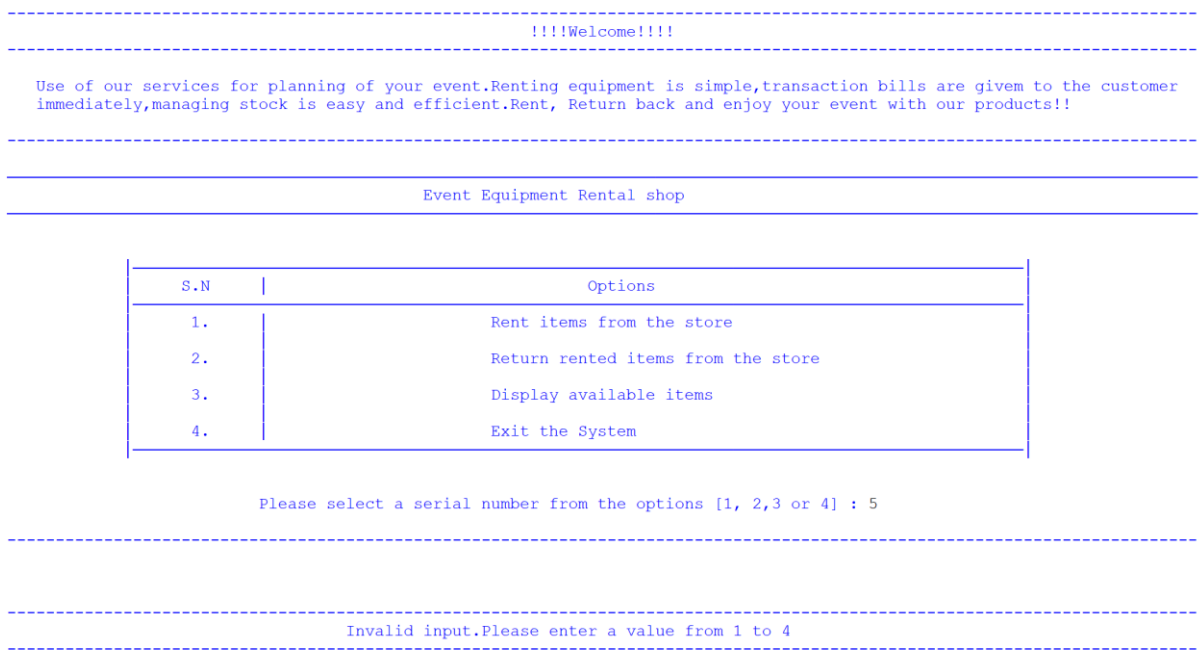


Figure 23: Invalid input

7.2) Test 2

Objective	Negative and non existed value is input by the user
Action	Negative and non existed value is input buy the user and error message for each is displayed
Expected Result	Invalid message should be displayed for both
Actual Result	Invalid message is displayed for both
Conclusion	Test was successful

Table 3: Test 2

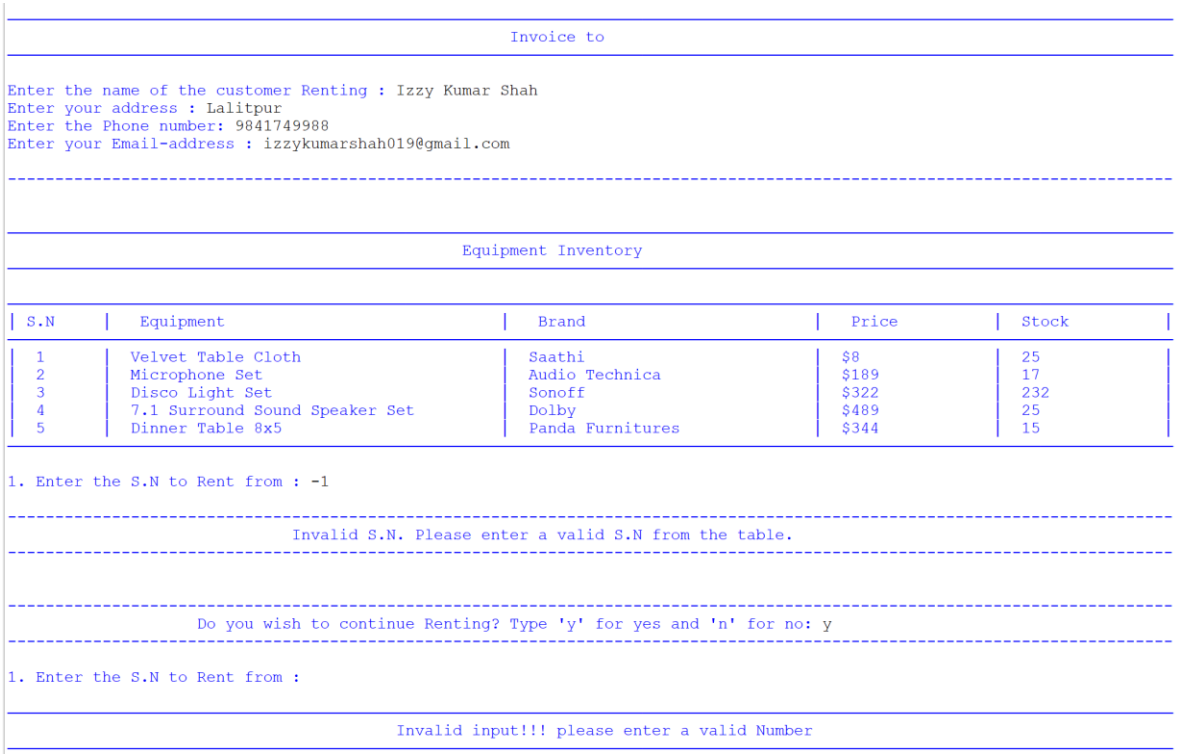


Figure 24:negative and non existed value

7.3) Test 3

Objective	To show complete process of file generation in purchase process when renting multiple items, Showing output in the shell ,showing rented items invoice details in a textfile
Action	purchase process when renting multiple items is shown, output is displayed in the shell , rented items invoice details is shown in a textfile
Expected Result	All the complete purchase should be with output in a shell and rented items in a invoice text file
Actual Result	All the complete purchase is shown with output in a shell and rented items in a invoice text file
Conclusion	Test was successful

Table 4: Test 3

Showing complete purchase process when renting multiple items with output in shell

```
------------------------------------------------------------------------------------------------------------------------
                                                !!!!Welcome!!!!
------------------------------------------------------------------------------------------------------------------------

   Use of our services for planning of your event.Renting equipment is simple,transaction bills are givem to the customer
   immediately,managing stock is easy and efficient.Rent, Return back and enjoy your event with our products!!

------------------------------------------------------------------------------------------------------------------------

                                            Event Equipment Rental shop

|-------------------------------------------------------------------------------|
|    S.N    |                              Options                              |
|-------------------------------------------------------------------------------|
|     1.    |                    Rent items from the store                      |
|     2.    |                    Return rented items from the store             |
|     3.    |                    Display available items                        |
|     4.    |                    Exit the System                                |
|-------------------------------------------------------------------------------|

                    Please select a serial number from the options [1, 2,3 or 4] : 1

------------------------------------------------------------------------------------------------------------------------

                                                Invoice to

Enter the name of the customer Renting : Izzy Kumar Shah
Enter your address : Lalitpur
Enter the Phone number: 9841749988
Enter your Email-address : izzykumarshah019@gmail.com

------------------------------------------------------------------------------------------------------------------------

                                            Equipment Inventory

| S.N |  Equipment                  |  Brand             |  Price  |  Stock  |
|-----|-----------------------------|--------------------|---------|---------|
|  1  |  Velvet Table Cloth         |  Saathi            |  $8     |  25     |
|  2  |  Microphone Set             |  Audio Technica    |  $189   |  17     |
|  3  |  Disco Light Set            |  Sonoff            |  $322   |  232    |
|  4  |  7.1 Surround Sound Speaker Set |  Dolby         |  $489   |  25     |
|  5  |  Dinner Table 8x5           |  Panda Furnitures  |  $344   |  15     |

1. Enter the S.N to Rent from : 2
2. Enter the quantity to Rent : 4

------------------------------------------------------------------------------------------------------------------------
                    Do you wish to continue Renting? Type 'y' for yes and 'n' for no: y
------------------------------------------------------------------------------------------------------------------------

1. Enter the S.N to Rent from : 3
2. Enter the quantity to Rent : 6

------------------------------------------------------------------------------------------------------------------------
                    Do you wish to continue Renting? Type 'y' for yes and 'n' for no: n
------------------------------------------------------------------------------------------------------------------------

                                        Your Invoice has been generated
```

Figure 25: Complete renting process with output in shell

Showing rented items invoice details in a text file

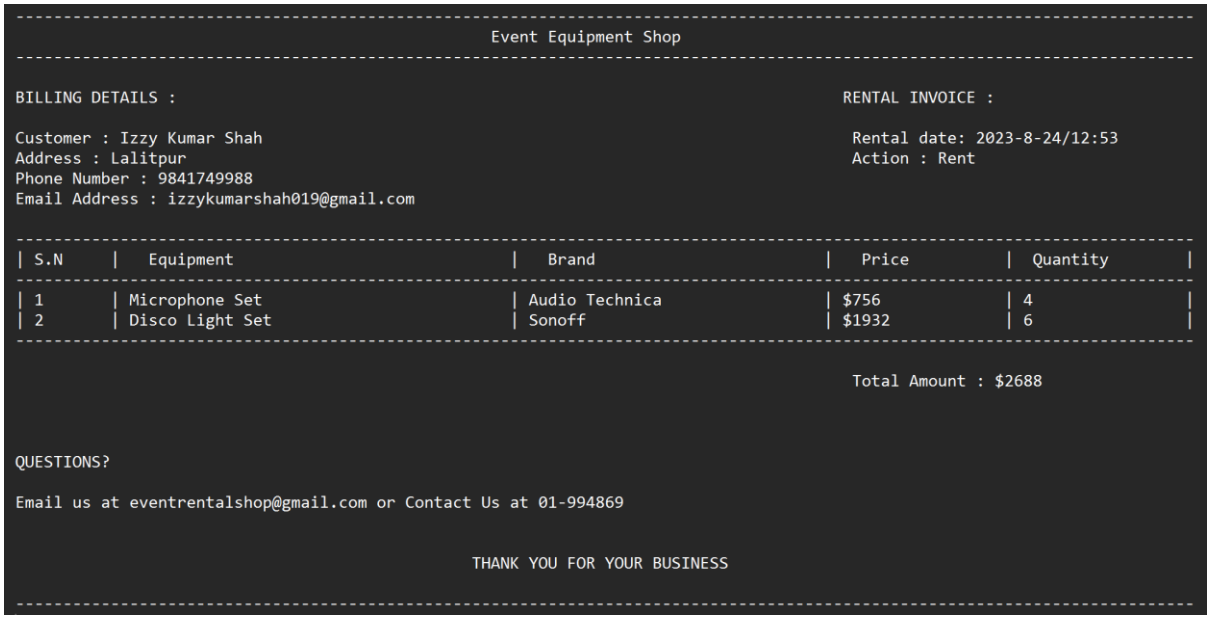


Figure 26: Rental invoice

7.3) Test 4

Objective	To show complete process of file generation when renturning multiple items, Showing output in the shell, showing renturned items invoice details in a textfile
Action	purchase process when renturning multiple items is shown, output is displayed in the shell, returned items invoice details is shown in a textfile
Expected Result	All the complete purchase should be with output in a shell and returned items in a invoice text file
Actual Result	All the complete purchase is shown with output in a shell and renturned items in a invoice text file
Conclusion	Test was successful

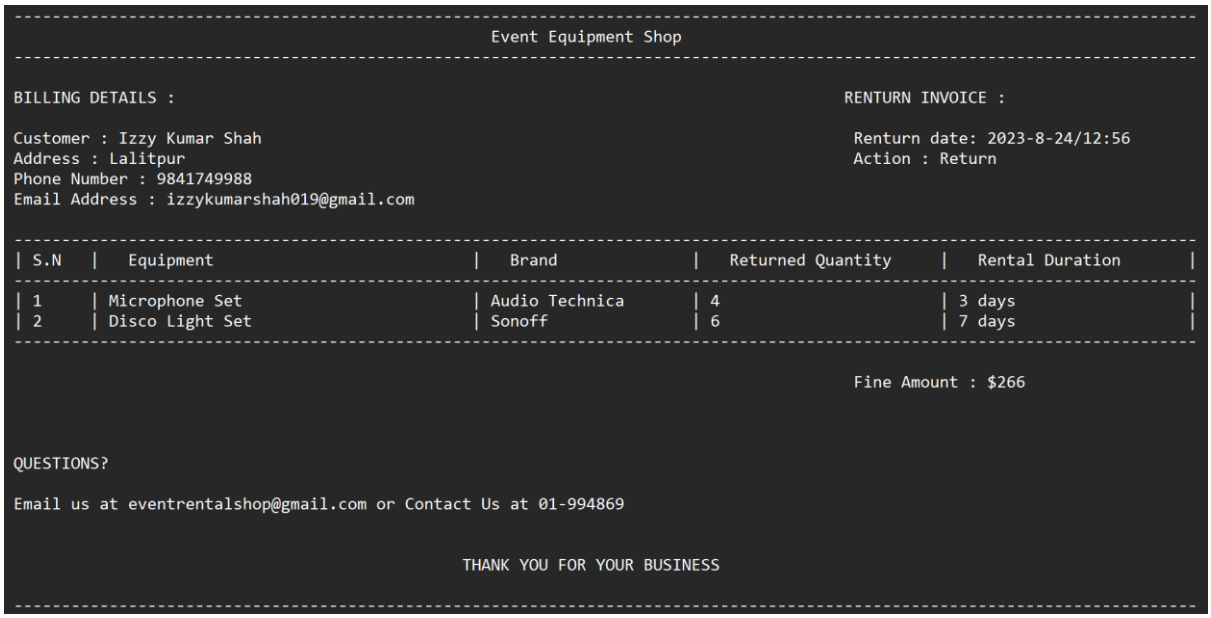
Table 5: Test 4

Showing complete purchase process when renturning multiple items with output in shell

```
Event Equipment Rental shop

| S.N | Options |
| 1.  | Rent items from the store |
| 2.  | Return rented items from the store |
| 3.  | Display available items |
| 4.  | Exit the System |

Please select a serial number from the options [1, 2,3 or 4] : 2
------------------------------------------------------------------------------------------------------------

Invoice to

Enter the name of the customer Returning : Izzy Kumar Shah
Enter your address : Lalitpur
Enter the Phone number: 9841749988
Enter your Email-address : izzykumarshah019@gmail.com
------------------------------------------------------------------------------------------------------------

Equipment Inventory

| S.N | Equipment                    | Brand            | Price | Stock |
| 1   | Velvet Table Cloth           | Saathi           | $8    | 25    |
| 2   | Microphone Set               | Audio Technica   | $189  | 13    |
| 3   | Disco Light Set              | Sonoff           | $322  | 226   |
| 4   | 7.1 Surround Sound Speaker Set | Dolby          | $489  | 25    |
| 5   | Dinner Table 8x5             | Panda Furnitures | $344  | 15    |

1. Enter the S.N to Return from : 2
2. Enter the quantity to Return : 4
3. Please enter Your rental duration [in days]: 3
------------------------------------------------------------------------------------------------------------
Do you wish to continue Returning? Type 'y' for yes and 'n' for no: y

1. Enter the S.N to Return from : 3
2. Enter the quantity to Return : 6
3. Please enter Your rental duration [in days]: 7
------------------------------------------------------------------------------------------------------------
Do you wish to continue Returning? Type 'y' for yes and 'n' for no: n

Your Invoice has been generated
------------------------------------------------------------------------------------------------------------
```

Figure 27: Complete renturning process with output in shell

Showing renturning items invoice details in a text file



```
-------------------------------------------------------------------------------------------------------------------
                                            Event Equipment Shop
-------------------------------------------------------------------------------------------------------------------

BILLING DETAILS :                                                                RENTURN INVOICE :

Customer : Izzy Kumar Shah                                                        Renturn date: 2023-8-24/12:56
Address : Lalitpur                                                                Action : Return
Phone Number : 9841749988
Email Address : izzykumarshah019@gmail.com

-------------------------------------------------------------------------------------------------------------------
| S.N   |   Equipment                       |   Brand             |   Returned Quantity     |   Rental Duration       |
-------------------------------------------------------------------------------------------------------------------
| 1     | Microphone Set                    | Audio Technica      | 4                       | 3 days                  |
| 2     | Disco Light Set                   | Sonoff              | 6                       | 7 days                  |
-------------------------------------------------------------------------------------------------------------------

                                                                                 Fine Amount : $266



QUESTIONS?

Email us at eventrentalshop@gmail.com or Contact Us at 01-994869

                                            THANK YOU FOR YOUR BUSINESS

-------------------------------------------------------------------------------------------------------------------
```

Figure 28: Return invoice

7.3) Test 5

Objective	To show updated stock of items after renting and returning in txt file compared with original
Action	Stock of items after renting and returning in txt file is updated compared with original
Expected Result	Stock should be update for rent and returned items. In the inventory file
Actual Result	Stock is updated for rent and returned items. In the inventory file
Conclusion	Test was successful

Table 6: Test 5

Original inventory.txt

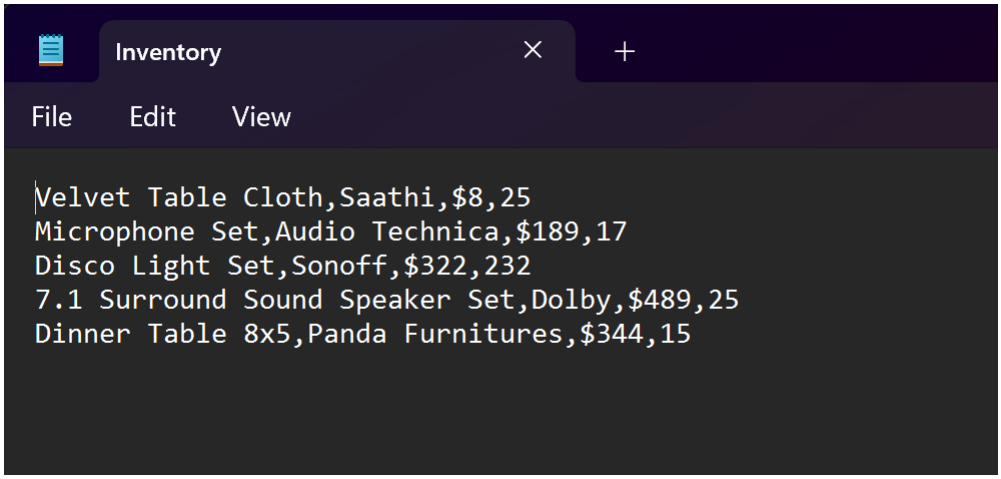


Figure 29: Original inventory.txt

After renting the updated inventory.txt

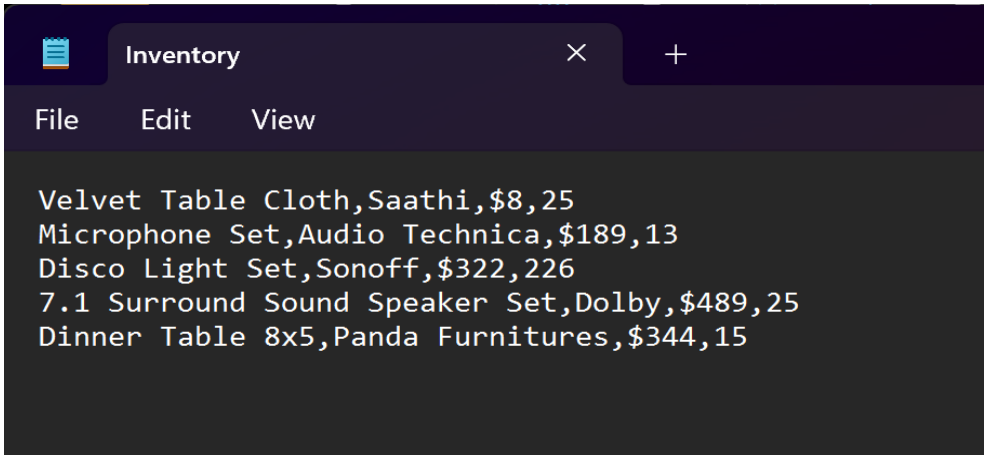


Figure 30:Updated after renting

After returning the rented items updated inventory.txt

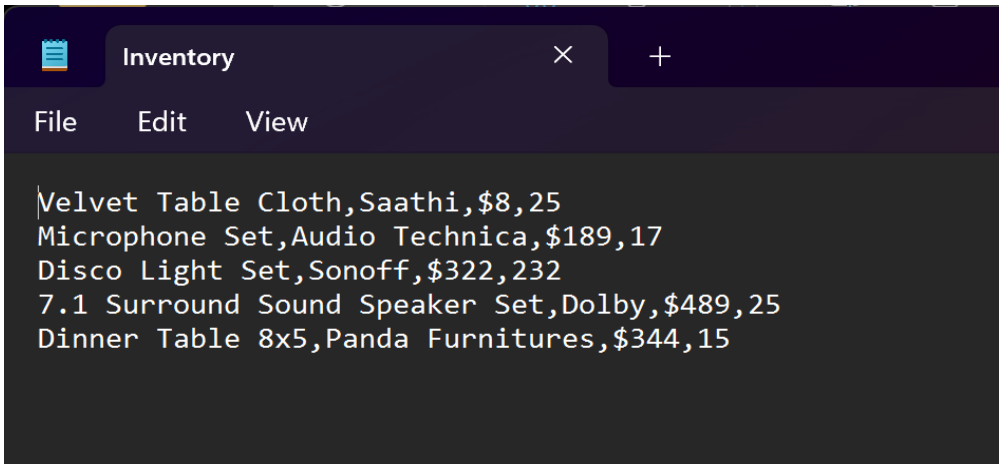


Figure 31:Updated after returning

8. Conclusion

This coursework of developing Event Equipment Rental shop application has been a great learning experience. I designed a system that handles renting, returning, generating invoice, updating and keeping stock information smoothly. Breaking the bigger and hard task into smaller parts help me during my creating of the application as I felt overwhelmed at first. It also helped me to find out where the error was exactly or where do I want to make the changes in the code. I made the interface of the application to look simple and user friendly. Writing algorithm and creating flowchart helped me understand how the program should work and it helped me visualize the flow of the code. During my coursework I faced many problems due to improper indentation. This coursework also helped me to use conditional statement properly. I learned using of try except for exception handling was important as when user inputs invalid input the program should not break instead it should show error message. During this coursework I properly learned about primitive and collection data types and the major use of collection data types as it stored information of the user. I also learned about docstring and implemented it in my code. Use of datetime module was also a new concept to me during the coursework as using that to store customer rent and return time and to make the file unique as using datetime makes file number unique. Iterating through loop also caused a lot of errors as i was looping wrong elements in the loop which gave me a different output as well as many syntax errors. Removing '$' sign during the process part and also adding after the data is processed and need to show it in a output was time consuming for me as i faced many errors and I modified my code a lot. Designing the invoice in a simple easy way was hard as managing the space between the lines where to make box and where to end a line was also quite hard. To make the app and invoice to look practical and real world usable I researched a lot of invoice design and what invoice are popular. During the coursework I learned how to create unique file, organizing, updating inventory, separating different invoices as using same invoices creates confusion to both user and admin. There are certain limitations like lack of GUI and database as a GUI would have made the app look better as there is no color buttons text area combobox which would have made the application easy to use and look better. Lack of database also causes the company to data loss if crash happens in the app. It is not safe as database because the information is only store in a text file. Storing equipment data, invoices, customer information would have been easy as storing it in database is easy and secure. Overall, I have really enjoyed working on this project and learned a lot about software development on how to design, build and make a fully working application which will help me in my future projects and working activities. The ideas and knowledge I learned during this coursework will help me make better programmer and help me make better decision.

References

BYJU'S, 2023. *BYJU'S.* [Online]
Available at: https://byjus.com/govt-exams/microsoft-word/
[Accessed 6 AUG 2023].

FTC, 2023. *FTC.* [Online]
Available at: https://fotc.com/blog/draw-io-online-guide/
[Accessed 6 aug 2023].

javatpoint, 2023. *javatpoint.* [Online]
Available at: https://www.javatpoint.com/idle-software-in-python#:~:text=IDLE%20stands%20for%20Integrated%20Development,IDLE%2C%20an%20integrated%20development%20environment.
[Accessed 15 aug 2023].

9. Appendix

9.1) Main.py

```
import ReadFile
import Operation
import WriteFile
import Message

def main():
    data = ReadFile.readData()
    Message.display_welcome_message()
    option=0

    while True:
        Message.display_option_message()
        try:
            option = int(input(" "*25+" Please select a serial number from the options [1, 2,3 or 4] : "))
        except ValueError:
            print("\n"+"---"*41)
```

```
        print(" "*41+"Invalid input!!! please enter a valid Number")
        print("---"*41+"\n")
    print("\n"+"---"*41+"\n"*2)
    if 0 <= option > 4:
        print("\n"+"---"*41)
        print(" "*35+"Invalid input.Please enter a value from 1 to 4")
        print("---"*41+"\n")
        continue

    if option == 1:
        print("——"*41)
        print(" "*53+"Invoice to")
        print("——"*41+"\n")
        customer_name = input("Enter the name of the customer Renting : ")
        address=input("Enter your address : ")
        try:
            phone_number=int(input("Enter the Phone number: "))
        except ValueError:
            print("\n"+"——"*41)
            print(" "*41+"Invalid input!!! please enter a valid Number")
            print("——"*41+"\n")
            continue
        email_address= input("Enter your Email-address : ")
        print("\n"+"---"*41+"\n"*2)
        Operation.displayData(data)
        total_amount = 0
        total_quantity = 0
        store = []

        while True:
```

```
            try:
                row = int(input("\n"+"1. Enter the S.N to Rent from : "))
            except ValueError:
                print("\n"+"——"*41)
                print(" "*41+"Invalid input!!! please enter a valid Number")
                print("——"*41+"\n")
                continue
            if 1 <= row <= len(data):
                try:
                    quantity = int(input("2. Enter the quantity to Rent : "))
                except ValueError:
                    print("\n"+"——"*41)
                    print(" "*41+"Invalid input!!! please enter a valid Number")
                    print("——"*41+"\n")
                    continue
                available_quantity = int(data[row - 1][3])
                if 1 <= quantity <= available_quantity:
                    store.append((row, quantity))
                    item = data[row - 1]
                    action = 'Rent'
                    total_amount += item[2] * quantity
                    total_quantity += quantity
                    rent_invoice = Message.generate_rental_invoice( customer_name,
action, total_amount, quantity, store, data,phone_number,address,email_address)
                    data = Operation.update_Data(data, row, quantity)
                    WriteFile.write_Transaction_rent(rent_invoice, customer_name)
                    WriteFile.write_Data(data)


                else:
                    print("——"*41)
                    print(" "*52+"Invalid Quantity")
```

```
                print("——"*41+"\n")
                print(" "*35+"Insufficient stock!!! Please enter a valid stock." + "\n" +" "*35+ "Available quantity:", available_quantity)
            else:
                print("\n"+"---"*41+"\n"+" "*30+"Invalid S.N. Please enter a valid S.N from the table."+"\n"+"---"*41+"\n")
            print("\n"+"---"*41)
            ask = input(" "*20+"Do you wish to continue Renting? Type 'y' for yes and 'n' for no: ")
            print("---"*41)
            if ask.lower() == 'n' or ask.lower() == 'no':
                print("\n"+"——"*41)
                print(" "*41+"Your Invoice has been generated")
                print("——"*41+"\n")
                break

    elif option == 2:
        returned_items = []
        returned_quantities = []
        fine_amounts = []
        rental_durations = []
        print("——"*41)
        print(" "*53+"Invoice to")
        print("——"*41+"\n")
        customer_name = input("Enter the name of the customer Returning : ")
        address=input("Enter your address : ")
        try:
            phone_number=int(input("Enter the Phone number: "))
        except ValueError:
            print("\n"+"——"*41)
            print(" "*41+"Invalid input!!! please enter a valid Number")
```

```
            print("——"*41+"\n")
            continue
        email_address= input("Enter your Email-address : ")
        print("\n"+"---"*41+"\n"*2)
        Operation.displayData(data)
        isTrue = True

        while isTrue:
            try:
                row = int(input("\n"+"1. Enter the S.N to Return from : "))
            except ValueError:
                print("\n"+"——"*41)
                print(" "*41+"Invalid input!!! please enter a valid Number")
                print("——"*41+"\n")
                continue
            if 1 <= row <= len(data):
                try:
                    returned_quantity = int(input("2. Enter the quantity to Return : "))
                except ValueError:
                    print("\n"+"——"*41)
                    print(" "*41+"Invalid input!!! please enter a valid Number")
                    print("——"*41+"\n")
                    continue
                try:
                    rental_duration = int(input("3. Please enter Your rental duration [in
days]: "))
                except ValueError:
                    print("\n"+"——"*41)
                    print(" "*41+"Invalid input!!! please enter a valid Number")
                    print("——"*41+"\n")
                    continue
```

```
                item = data[row - 1]
                returned_items.append((item, row))
                returned_quantities.append(returned_quantity)
                rental_durations.append(rental_duration)
                fine_per_day = int(item[2])


                rental_days = 5
                late_days = rental_duration - rental_days
                fine_amount = late_days * fine_per_day
                fine_amounts.append(fine_amount)
                action = 'Return'
                return_invoice = Message.generate_return_invoice(returned_items,
customer_name, action, returned_quantities, fine_amounts, phone_number,
rental_durations,address,email_address)
                data = Operation.update_Data(data, row, -returned_quantity)
                WriteFile.write_Transaction_return(return_invoice, customer_name)
                WriteFile.write_Data(data)
            else:
                print("\n"+"---"*41+"\n"+" "*30+"Invalid S.N. Please enter a valid S.N
from the table."+"\n"+"---"*41+"\n")
            print("\n"+"---"*41)
            ask = input(" "*20+"Do you wish to continue Renting? Type 'y' for yes and
'n' for no: ")
            print("---"*41)
            if ask.lower() == 'n' or ask.lower() == 'no':
                print("\n"+"——"*41)
                print(" "*41+"Your Invoice has been generated")
                print("——"*41+"\n")
                isTrue = False
                print("---"*41)


    elif option == 3:
```

```
            Operation.displayData(data)

        elif option == 4:
            Message.display_exit_message()
            break

main()
```

9.2) ReadFile.py

```
def readData():
    file = open("Inventory.txt", "r")
    data = []
    for line in file.readlines():
        row = line.replace("\n", "").split(",")
        row[2] = int(row[2].replace("$", ""))
        data.append(row)
    file.close()
    return data
```

9.3) WriteFile.py

```
import Operation

def write_Data(data):
    file = open("Inventory.txt", "w")
    for row in data:
        changed_row = []
        for index in range(len(row)):
            item = row[index]
```

```
            if index == 2:
                changed_row.append('$' + str(item))
            else:
                changed_row.append(str(item))
        file.write(",".join(changed_row) + "\n")
    file.close()


def write_Transaction_rent(rent_invoice, customer_name):
    times=str(Operation.file_time())
    transaction_file_name = customer_name+"_" +"Rent"+"_"+times+ "_"
+"transaction.txt"
    file = open(transaction_file_name, "w")
    file.write(rent_invoice + "\n")
    file.close()


def write_Transaction_return(return_invoice, customer_name):
    times=str(Operation.file_time())
    transaction_file_name =  customer_name+"_" +"Return"+"_"+times+ "_"
+"transaction.txt"
    file = open(transaction_file_name, "w")
    file.write(return_invoice + "\n")
    file.close()
```

9.4) Operation.py

```
import datetime


def displayData(data):
    print("——"*41)
    print(" "*48+"Equipment Inventory")
    print("——"*41+"\n")
```

```
    print("——"*41)
    print("│"+" "+"S.N"+" "*5+"│"+" "*3+"Equipment"+" "*29+"│"+" "*3+"Brand"+"
"*24+"│"+" "*3+"Price"+" "*10+"│"+" "*2+"Stock"+" "*10+"│")
    print("——"*41)
    for item_num in range(len(data)):
        row = data[item_num]
        sn = item_num + 1
        print("│  " + " " + str(sn) + " " * (7 - len(str(sn))) +
              "│  " + " " + row[0] + " " * (39 - len(row[0])) +
              "│  " + " " + row[1] + " " * (30 - len(row[1])) +
              "│  " + " $" + str(row[2]) + " " * (16 - len("$" + str(row[2]))) +
              "│  " + " " + row[3] + " " * (15 - len(row[3])) +"│")
    print("——"*41)


def update_Data(data, row, quantity):
    data[row - 1][3] = str(int(data[row - 1][3]) - quantity)
    return data

def function_time():
    year = str(datetime.datetime.now().year)
    month = str(datetime.datetime.now().month)
    day = str(datetime.datetime.now().day)
    hour = str(datetime.datetime.now().hour)
    minute = str(datetime.datetime.now().minute)

    final_time = year + "-" + month + "-" + day + "/" + hour + ":" + minute

    return final_time
```

```
def file_time():
    year = str(datetime.datetime.now().year)
    month = str(datetime.datetime.now().month)
    day = str(datetime.datetime.now().day)
    hour = str(datetime.datetime.now().hour)
    minute = str(datetime.datetime.now().minute)

    file_time = str(year  + month  + day  + hour  + minute )

    return file_time
```

9.5) Message.py

```
import Operation

def display_option_message():
    print("\n"+"——"*41)
    print(" "*43+"Event Equipment Rental shop")
    print("——"*41+"\n")
    print("""
```

```
|————————————————————————————————————————————————————————————————————————|
        |   S.N   |                    Options                          |
|————————————————————————————————————————————————————————————————————————|
        |    1.   |          Rent items from the store                  |
        |         |                                                     |
        |    2.   |          Return rented items from the store         |
        |         |                                                     |
        |    3.   |          Display available items                    |
```

```
        |          |                                                    |
        |    4.    |              Exit the System                       |
|————————————————————————————————————————————————————————————————————
————————————————————————————————————————————————————————————————————|

    """)


def display_welcome_message():
    print("\n"+"---"*41)
    print(" "*54+"!!!!Welcome!!!!")
    print("---"*41+"\n")
    print(" "*3+"Use of our services for planning of your event.Renting equipment is
simple,transaction bills are givem to the customer")
    print(" "*3+"immediately,managing stock is easy and efficient.Rent, Return back
and enjoy your event with our products!!")
    print("\n"+"---"*41)


def display_exit_message():
    print("\n"+"---"*41)
    print(" "*56+"!!!!Thank You!!!!")
    print("---"*41+"\n")
    print(" "*35+"Thank you for choosiing our services.Please visit Again")
    print("\n"+"---"*41)


def generate_rental_invoice(customer_name, action, total_amount, quantity, store,
data,phone_number,address,email_address):
    invoice = (
        "--"*62+"\n"+
        " "*50+"Event Equipment Shop\n"+
        "--"*62+"\n"+
        "\n"+"BILLING DETAILS : "+" "*69+"RENTAL INVOICE : "+"\n"*2+
```

```
        "Customer : " + customer_name + " "*62+ "Rental date: " +
Operation.function_time()+"\n"+
        "Address : "+str(address)+" "*70+"Action : " + action +"\n"
        "Phone Number : "+str(phone_number)+"\n"
        "Email Address : "+str(email_address)+"\n"*2 +
        "----"*31)
    invoice+=("\n"+"|"+" "+"S.N"+" "*5+"|"+" "*3+"Equipment"+" "*29+"|"+"
"*3+"Brand"+" "*24+"|"+" "*3+"Price"+" "*10+"|"+" "*2+"Quantity"+" "*8+"|"+"\n")
    invoice+="----"*31+"\n"

    sn = 1
    for items in store:
        row = items[0]
        quantity = items[1]
        item_info = data[row - 1]
        invoice += (
            "|" + " " + str(sn) + " " * (8 - len(str(sn))) +
            "|" + " " +  item_info[0] + " " * (40 - len(item_info[0])) +
            "|" + " " +  item_info[1] + " " * (31 - len(item_info[1])) +
            "|" + " " + "$" + str(item_info[2] * quantity) + " " * (17 - len("$" + str(item_info[2]
* quantity))) +
            "|" + " " + str(quantity) + " " * (16 - len(str(quantity))) + " " + "|\n"
        )
        sn += 1
    invoice+="----"*31+"\n"*2

    invoice += (
        "  "*44+"Total Amount : $" + str(total_amount) + "\n"*4+
        "QUESTIONS?"+"\n"*2+
        "Email us at eventrentalshop@gmail.com or Contact Us at 01-994869 "+"\n"*3+
        " "*48+"THANK YOU FOR YOUR BUSINESS"+"\n"*2+
```

```
        "--"*62
    )
    return invoice


def generate_return_invoice(items, customer_name, action, returned_quantities,
fine_amounts, phone_number, rental_durations,address,email_address):
    invoice = (
        "--"*62+"\n"+
        " "*50+"Event Equipment Shop\n"+
        "--"*62+"\n"+
        "\n"+"BILLING DETAILS : "+" "*69+"RENTURN INVOICE : "+"\n"*2+
        "Customer : " + customer_name + " "*62+ "Renturn date: " +
Operation.function_time()+"\n"+
        "Address : "+str(address)+" "*70+"Action : " + action +"\n"
        "Phone Number : "+str(phone_number)+"\n"
        "Email Address : "+str(email_address)+"\n"*2 +
        "----"*31)
    invoice += ("\n"+"|"+" "+"S.N"+" "*3+"|"+" "*3+"Equipment"+" "*27+"|"+"
"*3+"Brand"+" " *14+"|"+" "*3+"Returned Quantity"+" "*5+"|"+" "*3+"Rental
Duration"+" "*7 +"|" + "\n")
    invoice += "----"*31 + "\n"


    total_fine = 0
    for index in range(len(items)):
        item, row = items[index]
        returned_quantity = returned_quantities[index]
        fine_amount = fine_amounts[index]
        total_fine += fine_amount
        rental_duration = rental_durations[index]
        invoice += (
            "|" + " " + str(index + 1) + " " * (6 - len(str(index + 1))) +
            "|" + " " + item[0] + " " * (38 - len(item[0])) +
```

```
            "|" + " " + item[1] + " " * (21 - len(item[1])) +
            "|" + " " + str(returned_quantity) + " " * (24 - len(str(returned_quantity))) +
            "|" + " " + str(rental_duration) + " days" + " " * (19 - len(str(rental_duration))) +
"|" + "\n"
        )
    invoice += "----"*31 + "\n"*2


    if total_fine > 0:
        invoice += (
            "  "*44+"Fine Amount : $" + str(total_fine) + " " * (17 - len(str(total_fine))) +
"\n"*4+
            "QUESTIONS?"+"\n"*2+
            "Email us at eventrentalshop@gmail.com or Contact Us at 01-994869
"+"\n"*3+
            " "*47+"THANK YOU FOR YOUR BUSINESS"+"\n"*2+
            "--"*62
        )
    else:
        invoice += "  "*44+"Fine Amount : $0" + " " * 14 +
"\n"*4+"QUESTIONS?"+"\n"*2+"Email us at eventrentalshop@gmail.com or Contact
Us at 01-994869 "+"\n"*3+" "*48+"THANK YOU FOR YOUR BUSINESS"+"\n"*2+ "--
"*62


    return invoice
```

9.6) Inventory.txt

```
Velvet Table Cloth,Saathi,$8,29
Microphone Set,Audio Technica,$189,17
Disco Light Set,Sonoff,$322,232
7.1 Surround Sound Speaker Set,Dolby,$489,25
Dinner Table 8x5,Panda Furnitures,$344,15
```