Islington College

Information System

CC4002NA

Coursework 2

Submitted By:

Student Name: Abiral Joshi-Shrestha

LondonMet ID: NP01NT4A170008

Group: L1N1

Date: 12th Jan'2018

Submitted To:

Mr. Sukrit Shakya

Module Leader

Information System

Contents

1. Proposal

My project is a basic inventory management system for an electronics store to manage and maintain its inventory. The application reads the text file and displays the products available in the inventory. It also generates an invoice for each purchase for the particular customer and also writes that into a file. In addition to that, it also updates the inventory with each purchase.

As of the current time and day, most retail stores use a basic method for managing and updating the inventory and to keep track of the customers' purchases. Most retail stores currently use paper bills and written ledgers to note and manage all the products in their inventory and to keep track of the stocks that have been sold. This method of keeping track of the activities which are carried out in the store as the information could be easily destroyed, stole or lost. So to store, modify and update the information in such a manner was a very time consuming, inefficient and laborious task.

The main aim and objective for creating this Inventory Management System is to increase the efficiency in keeping track of the inventory, the stock and the customers' purchases of the electronic store. This system is also meant to reduce the manpower required to keep track of the inventory and the customers' purchases. The main issue in creating this system was to make it as efficient as possible. This was to reduce the processing time of the system. The other issue was to make the system simple and transparent so that even people with a basic knowledge of the system could use it easily.

So to resolve the issues of the electronics store, I have created a simple, yet effective, system. As most people have basic idea of a computer system, this system will greatly improve the working efficiency of the electronics store. The system will record the purchases of the customers, generate an invoice and additionally, it will update and keep track of the remaining goods in the inventory.

This system can be useful to all types of commercial uses, and for educational uses as well. This system is primarily aimed to be used by retail stores to manage their business. In educational field, it can be used to show new learners how a basic program can be created using Python.

The basic requirement to run this system is that the user must have a computer system with a suitable operating system. If the user has specific needs, the system can be modified accordingly.

2. Introduction

This is the report of the coursework which had provided the task of writing a program in Python that creates an application which will read the text file, in which the inventory is maintained, and displays the products available. Then with each purchase an invoice should be generated for the particular customer and should be written into the text file. The inventory should also be updated with each purchase. Furthermore, the program was needed to be developed with the help of an algorithm, a flowchart and pseudo code.

The tasks assigned which were to be done by us in this coursework was not easy. Writing such a complex and long program which performed such multiple tasks and writing the program such that it could be easily used by everyone, was a really difficult task. Thankfully, realizing the difficulty of the task in hand, this project was done with the utilization of all resources possible and with great determination. Since this project had multiple tasks and to do it without any planning would make the project very tiresome and unmanaged, some objectives were set while doing this projects. Some of these objectives are listed below:

- Separate the program into modules so as to make the program easier to understand and to write.
- Write pseudo code and algorithm of the program that manages, maintains and updates the inventory of an Electronics Store.
- Write a simple and user-friendly program making the use of the created algorithms, pseudo code and flowchart.
- Obtain an accurate and interactive program by carrying out a series of multiple tests.
- Debug the program if any errors occur.

In order to meet the objectives of this project, a lot of time, research and guidance was needed so as to make this coursework as effective and efficient as possible. I personally carried out a lot of research using various web sites, journals and books as reference. My module leader and my fellow class mates helped me in removing the doubts and confusions which I had in

my mind. So, I would also like to thank them for supporting me in the completion of this coursework and report.

The program for this report was created using Python programming software IDLE. The reason behind using IDLE is that it is very easy to use and also would help to make the program more efficient through the use of various in-built functions in it.

3. Discussion and Analysis

The program was developed using the Python programming language and was coded in the development environment “Integrated Development and Learning Environment” a.k.a. IDLE. I designed my program to make it simple and easy to use by people who have basic knowledge of computer systems. I have designed the program such that it can handle and resolve the errors given by the users and then ask the user to correct that error.

I have based my program upon various functions and modules to make it simple and easy to code. By creating functions, I made it easier to reuse the codes that are repeatedly used in the program. I created the program into four modules so that the important codes can be separated, this makes it a lot easier to debug the program if any problems or errors occur in the source code.

The technologies I used while making this are: basic computer system, internet access and Python programing language. To code the program, I used IDLE as it is easier to code in IDLE. The tools I have used to develop this program are the basic tools which come as a standard when anyone installs IDLE. Additionally, I store the items and products of the store in a Text document which I created using Notepad.

4. Algorithm

The required algorithm is as given below:

Step 1: Start.

Step 2: Import the modules into the main module.

Step 3: Create lists for each unique headers.

Step 4: Append the items of the text file into new lists.

Step 5: Open the file and split the items into separate lines.

Step 6: Define functions, Main and file.

Step 7: Introduce a loop and call menu for choice.

Step 8: If choice is equal to or entered as one then enter the loop.

Step 9: Ask the name of the product.

Step 10: Product quantity is verified.

Step 11: If the name of the product is invalid, step 10 is repeated until a valid answer is obtained.

Step 12: Product quantity is asked.

Step 13: If the entered value of quantity is less than 0 or greater than the available stock, it goes back to step 12.

Step 14: If the value of quantity entered is valid, the program enters into a new loop.

Step 15: The quantity and rate of product is calculated.

Step 16: Ask discount amount.

Step 17: If the discount amount is greater than the original price of no discount is given, step 16 is repeated.

Step 18: The final price is calculated according to the input of the user.

Step 19: Name of the customer is asked and the purchase is confirmed.

Step 20: The value of quantity is also sent to the module containing main.

Step 21: But if purchase is cancelled, the module main is left as it is.

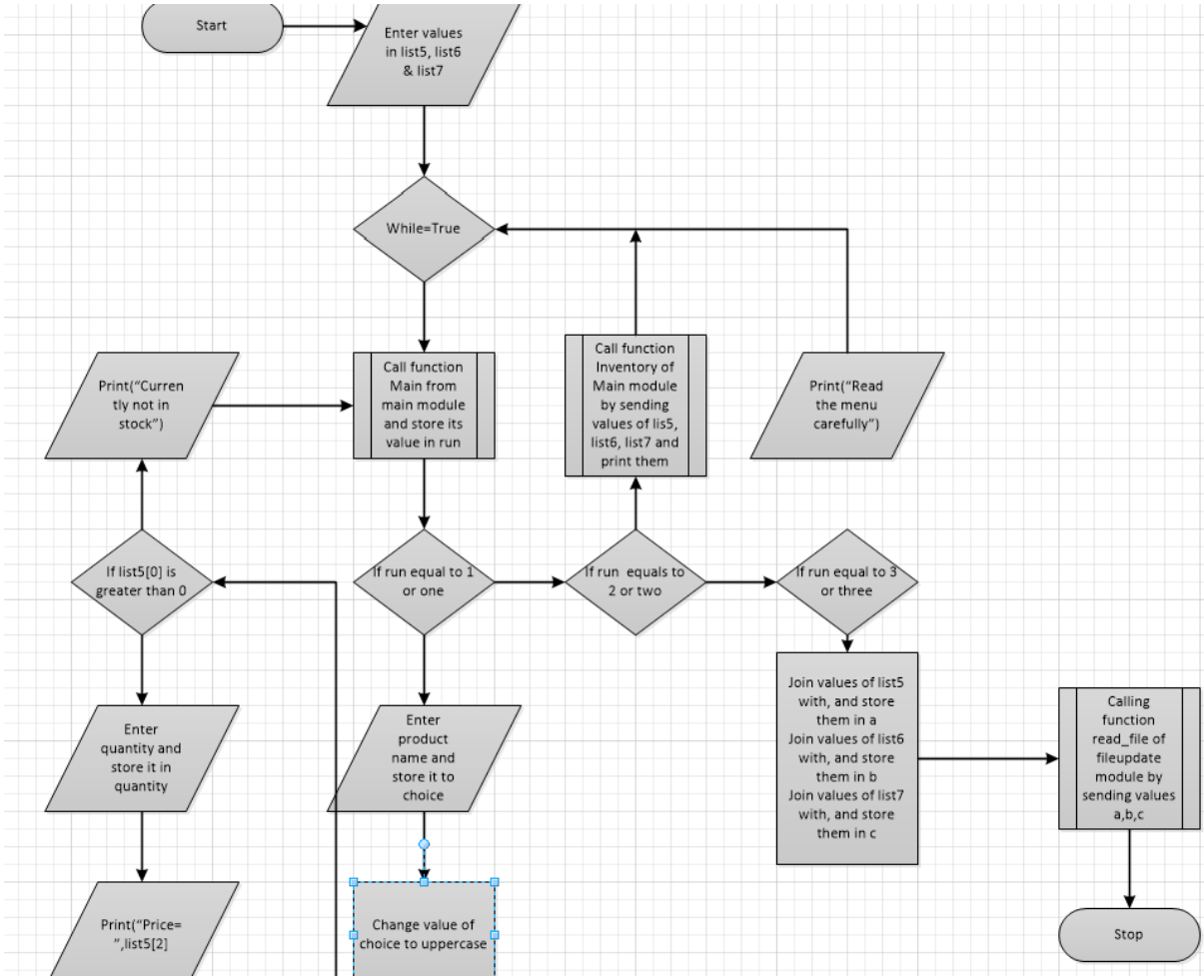
Step 22: The input taken from the customer is displayed in a next text file.

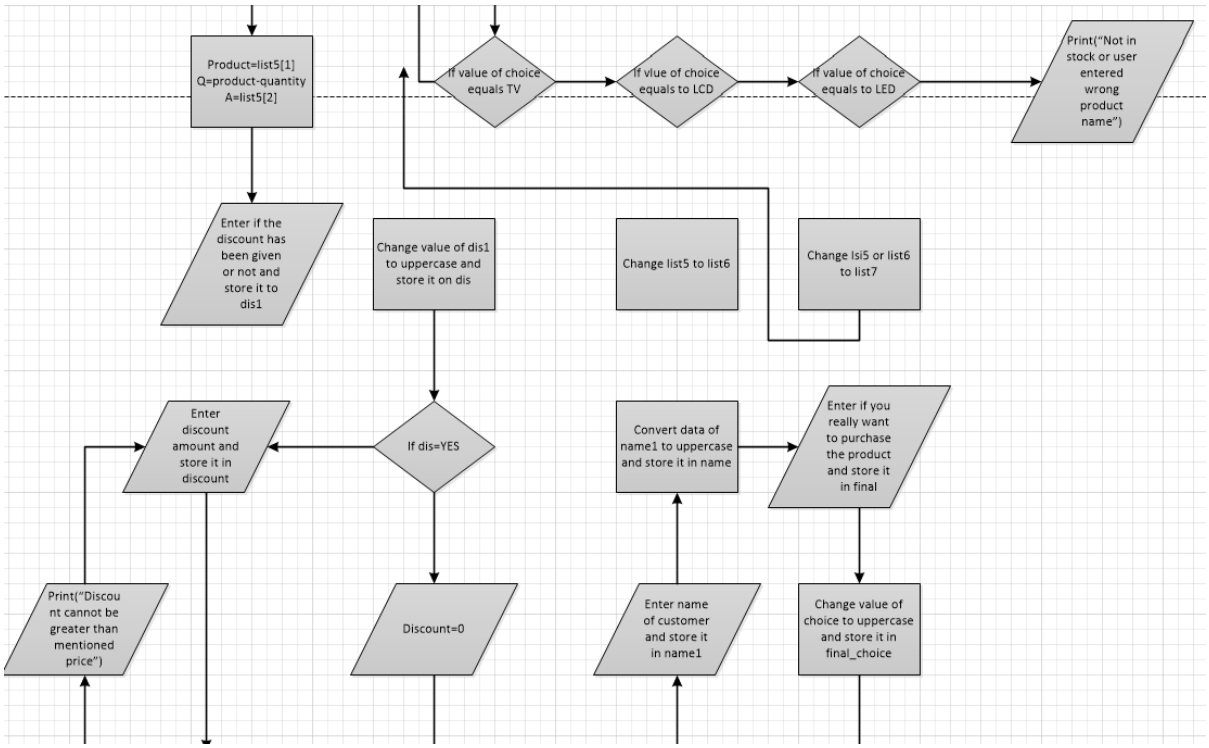
Step 23: If the number of products in the inventory drops to 0, a message “Out of Stock” is displayed.

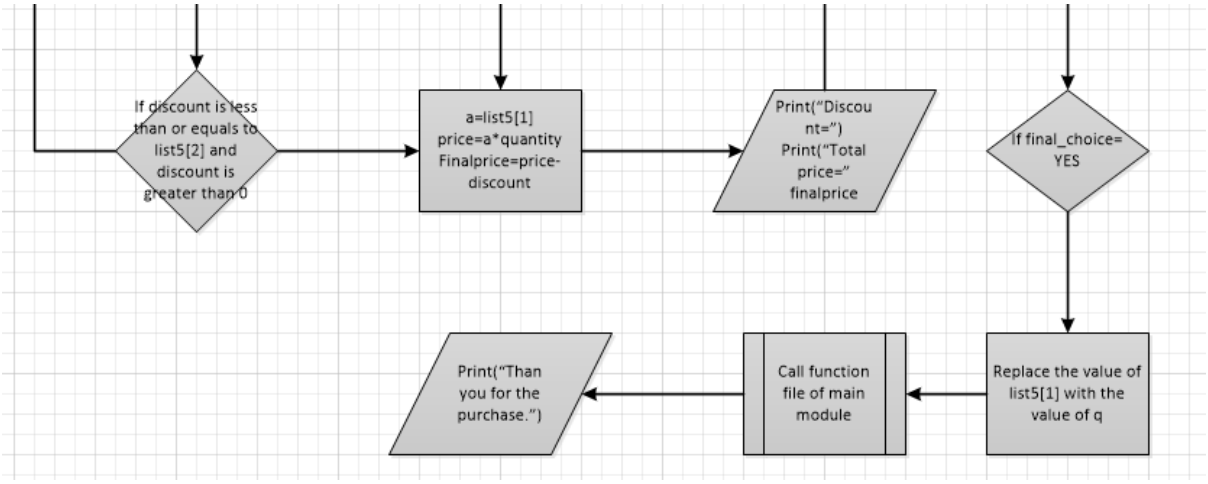
Step 34: End of program.

5. Pseudo code

6. Flowchart


Start
Enter values in list5, list6 & list7
While=True
Print("Currently not in stock")
Call function Main from main module and store its value in run
Call function Inventory of Main module by sending values of lis5, list6, list7 and print them
Print("Read the menu carefully")
If list5[0] is greater than 0
If run equal to 1 or one
If run equals to 2 or two
If run equal to 3 or three
Enter quantity and store it in quantity
Enter product name and store it to choice
Join values of list5 with, and store them in a
Join values of list6 with, and store them in b
Join values of list7 with, and store them in c
Calling function read_file of fileupdate module by sending values a,b,c
Print("Price=",list5[2]
Change value of choice to uppercase
Stop


Product=list5[1]
Q=product-quantity
A=list5[2]
If value of choice equals TV
If vlue of choice equals to LCD
If value of choice equals to LED
Print("Not in stock or user entered wrong product name")
Enter if the discount has been given or not and store it to dis1
Change value of dis1 to uppercase and store it on dis
Change list5 to list6
Change lsi5 or list6 to list7
Enter discount amount and store it in discount
If dis=YES
Convert data of name1 to uppercase and store it in name
Enter if you really want to purchase the product and store it in final
Print("Discount cannot be greater than mentioned price")
Discount=0
Enter name of customer and store it in name1
Change value of choice to uppercase and store it in final_choice


If discount is less than or equals to list5[2] and discount is greater than 0
a=list5[1]
price=a*quantity
Finalprice=price-discount
Print("Discou nt=")
Print("Total price="
finalprice
If final_choice= YES
Replace the value of list5[1] with the value of q
Call function file of main module
Print("Than you for the purchase.")

7. Program

This is the main module of the program. All the other modules are imported in to this modules and the processing occurs.

```
#Importing various modules
import Write
import Test
import Main

#Importing inbuilt functions
import random
import datetime

#Creating lists
list1=[]
list2=[]
list3=[]
list4=[]
list5=[]
list6=[]
list7=[]


file=open("Inventory.txt","r") #opening file for reading data
for line in file:
    x=line.split("\n")
    a=x[0]
    list1.append(a)
file.close()

list2.append(list1[0])
list3.append(list1[1])
list4.append(list1[2])


for i in list2: #appending various data to different lists
    x=i.split(",")
    a=x[0]
    b=x[1]
    c=x[2]
    list5.append(a)
    list5.append(int(b))
    list5.append(int(c))

for i in list3: #appending various data to different lists
    x=i.split(",")
    a=x[0]
    b=x[1]
    c=x[2]
    list6.append(a)
    list6.append(int(b))
```

```
    list6.append(int(c))

for i in list4: #appending various data to different lists
    x=i.split(",")
    a=x[0]
    b=x[1]
    c=x[2]
    list7.append(a)
    list7.append(int(b))
    list7.append(int(c))


def menu(): #Creating a function to show menu
    print("Press 1 to purchase a product")
    print("Press 2 to see available stock")
    print("Press 3 to exit")
    return input("Enter your choice=")


def file(): #Creating a function to write a file
    a=("customer"+str(random.randint(1,100)))
    d=str(datetime.datetime.now())
    Write.file(a,name,choice,d,finalprice,discount,quantity)


#Using while in order to keep the programing running until users decides to quit
print(Main.inventory(list5,list6,list7))
while True:
    run=menu()
    if run=="1" or run=="one":
        choice1=input("Enter product you want to purchase=")
        choice=choice1.upper() #Calling a module to convert lowercase string to uppercase string
        if choice=="TV":
            if list5[1]>0:
                #Using exception handling to take correct input from the user
                success=False
                while success==False:
                    try:
                        quantity=int(input("Enter quantity="))
                        if quantity<=list5[1] and quantity>0:
                            success=True
                        else:
                            print("------------------------------------------------------------")
                            print("Enter quantity is greater than or less than the available stock")
                            print("------------------------------------------------------------")
                    except:
                        print("------------------------------")
```

```
            print("------------------------------")
            print("Please enter quantity in numbers")
            print("------------------------------")

print("Price=",list5[2])
product=list5[1]
q=product-quantity #Calling a module to manage quantity
print("Quantity=",quantity)
a=list5[2]
sl=False
while sl==False:
    disl=input("Has discount been given(yes/no)?")
    dis=disl.upper()
    if dis=="YES":
        #Using exception handling to take correct input from the user
        success=False
        while success==False:
            try:
                discount=int(input("Enter discount amount="))
                if discount<=list5[2] and discount>0:
                    success=True
                else:
                    print("------------------------------------------------------------------------")
                    print("Discount amount cannot be greater than or less than the mentioned price")
                    print("------------------------------------------------------------------------")
            except:
                print("--------------------------------------")
                print("Please enter discount amount in numbers")
                print("--------------------------------------")
        sl=True

    elif dis=="NO":
        discount=0
        sl=True

    else:
        print("error")
        sl=False

price=a*quantity #Calling a module to find price by multiplying price with no of quantity
finalprice=price-discount #Calling a module to find toatl price with discount
print("Discount amount=",discount)
print("Total Price=",finalprice)
name1=input("Enter your name=")
name=name1.upper() #Calling a module to convert lowercase string to uppercase string
```

```
            #Using exception handling to take correct input from the user
            s2=False
            while s2==False:
                try:
                    final=input("Do you really want to purchase this product(yes/no)=")
                    final_choice=final.upper()
                    if final_choice=="YES":
                        list5[1]=q
                        file()
                        print("Thank you for your purchase")
                        s2=True
                    elif final_choice=="NO":
                        print(menus.stock(list5,list6,list7))
                        s2=True
                except:
                    print("Error")
        else:
            print("---------------------")
            print("Currently not in stock")
            print("---------------------")

    elif choice=="LCD":
        if list6[1]>0:
            #Using exception handling to take correct input from the user
            success=False
            while success==False:
                try:
                    quantity=int(input("Enter quantity="))
                    if quantity<=list6[1] and quantity>0:
                        success=True
                    else:
                        print("------------------------------------------------------------")
                        print("Enter quantity is greater than or less than the available stock")
                        print("------------------------------------------------------------")
                except:
                    print("------------------------------")
                    print("Please enter quantity in numbers")
                    print("------------------------------")

            print("Price=",list6[2])
            product=list6[1]
            q=product-quantity#Calling a module to manage quantity
            print("Quantity=",quantity)
            a=list5[2]
```

```
#Using exception handling to take correct input from the user
sl=False
while sl==False:
    disl=input("Has discount been given(yes/no)?")
    dis=disl.upper()
    if dis=="YES":
        #Using exception handling to take correct input from the user
        success=False
        while success==False:
            try:
                discount=int(input("Enter discount amount="))
                if discount<=list6[2] and discount>0:
                    success=True
                else:
                    print("------------------------------------------------------")
                    print("Discount amount cannot be greater than the mentioned price")
                    print("------------------------------------------------------")
            except:
                print("--------------------------------------")
                print("Please enter discount amount in numbers")
                print("--------------------------------------")
        sl=True

    elif dis=="NO":
        discount=0
        sl=True

    else:
        print("error")
        sl=False

price=a*quantity
finalprice=price-discount
print("Discount amount=",discount)
print("Total Price=",finalprice)
name1=input("Enter your name=")
name=name1.upper()

#Using exception handling to take correct input from the user
s2=False
while s2==False:
    try:
        final=input("Do you really want to purchase this product(yes/no)=")
        final_choice=final.upper()
        if final_choice=="YES":
            list6[1]=q
            file()
```

```
                    file()
                    print("Thank you for your purchase")
                    s2=True
                elif final_choice=="NO":
                    print(Main.inventory(list5,list6,list7))
                    s2=True
            except:
                print("Error")
    else:
        print("----------------------")
        print("Currently not in stock")
        print("----------------------")

elif choice=="LED":
    if list7[1]>0:
        success=False
        while success==False:
            try:
                quantity=int(input("Enter quantity="))
                if quantity<=list7[1] and quantity>0:
                    success=True
                else:
                    print("------------------------------------------------------------")
                    print("Enter quantity is greater than or less than the available stock")
                    print("------------------------------------------------------------")
            except:
                print("------------------------------")
                print("Please enter quantity in numbers")
                print("------------------------------")

        print("Price=",list7[2])
        product=list6[1]
        q=product-quantity
        print("Quantity=",quantity)
        a=list7[2]
        s1=False
        while s1==False:
            dis1=input("Has discount been given(yes/no)?")
            dis=dis1.upper()
            if dis=="YES":
                #Using exception handling to take correct input from the user
                success=False
                while success==False:
                    try:
                        discount=int(input("Enter discount amount="))
                        if discount<=list7[2] and discount>0:
```

```
                    success=True
                else:
                    print("-------------------------------------------------------------------------")
                    print("Discount amount cannot be greater than or less than the mentioned price")
                    print("-------------------------------------------------------------------------")
            except:
                print("--------------------------------------")
                print("Please enter discount amount in numbers")
                print("--------------------------------------")
            sl=True

        elif dis=="NO":
            discount=0
            sl=True

        else:
            print("error")
            sl=False

    price=a*quantity
    finalprice=price-discount
    print("Discount amount=",discount)
    print("Total Price=",finalprice)
    name1=input("Enter your name=")
    name=name1.upper()

    s2=False
    while s2==False:
        try:
            final=input("Do you really want to purchase this product(yes/no)=")
            final_choice=final.upper()
            if final_choice=="YES":
                list7[1]=q
                file()
                print("Thank you for your purchase")
                s2=True
            elif final_choice=="NO":
                print(Main.invertory
                      (list5,list6,list7))
                s2=True
        except:
            print("Error")
else:
    print("----------------------")
    print("Currently not in stock")
    print("----------------------")


    else:
        print("-------------------------------------------------")
        print("NOT IN STOCK OR USER ENTERED WRONG PRODUCT NAME")
        print("-------------------------------------------------")

elif run=="2" or run=="two":
    print(Main.inventory(list5,list6,list7))

elif run=="3" or run=="three":
    a=",".join([str(i) for i in list5]) #Joins elemnts of list with" , "
    b=",".join([str(i) for i in list6]) #Joins elemnts of list with" , "
    c=",".join([str(i) for i in list7]) #Joins elemnts of list with" , "

    Test.read_file(a,b,c) #Calling a module to overwrite the main stock file
    break

else:
    print("------------------------------")
    print("Please read the MENU carefully")
    print("------------------------------")
```

The rest of the codes are from the sub modules. They were used to simplify the program:

```
def inventory(list5, list6, list7):
    print("####")
    print("Product"+" "+"Quantity"+" "+"Price")
    print(list5[0]+"\t",str(list5[1])+"\t",str(list5[2])+"\t")
    print(list6[0]+"\t",str(list6[1])+"\t",str(list6[2])+"\t")
    print(list7[0]+"\t",str(list7[1])+"\t",str(list7[2])+"\t")
    print("####")
```

This is the coding used for the creation of the lists.

```
def read_file(a,b,c):
    file=open("Inventory.txt","w")
    file.write(str(a))
    file.write(str("\n"))
    file.write(str(b))
    file.write(str("\n"))
    file.write(str(c))
    file.write(str("\n"))
    file.close()
```

```
def file(a,name,choice,date,finalprice,discount,quantity):
    file=open(a+".txt","w")
    file.write(str("\n"))
    file.write(str("INVOICE"))
    file.write(str("\n"))
    file.write(str("INVOICE NO." + a))
    file.write(str("\n"))
    file.write(str("Name of Customer " + name))
    file.write(str("\n"))
    file.write(str("Product choosen " + choice))
    file.write(str("\n"))
    file.write(str("Date of purchase " + date))
    file.write(str("\n"))
    file.write("Price of product " + str(finalprice))
    file.write(str("\n"))
    file.write("Discount " + str(discount))
    file.write(str("\n"))
    file.write("Quantity " + str(quantity))
    file.write(str("\n"))
    file.write(str("INVOICE"))
    file.close()

    print("YOUR INVOICE")
    print("INVOICE NO.", a)
    print("Name of Customer ", name)
    print("Product choosen ", choice)
    print("Date of purchase ", date)
    print("Price of product ", finalprice)
    print("Discount ", discount)
    print("Quantity ", quantity)
    print("YOUR INVOICE")
```

This is the coding to open and read the text file, Inventory.

8. Data Structures

The different data types used in the program are:

A) String

String is a data type in python which is used to store a textual data. They help us to store a sequence of one or more characters that can be either a constant or a variable. In python, string values are enclosed within single or double quotes, the quotes tell python where the string begins and ends.

Syntax: <Variable name>=string value

String is used in the program to input and output string characters. Strings are mostly used to ask for the string information from the user.

B) Integer

Integer is a data type in python which stores the integer value. Any numeric value can be assigned to it at any given time, thus overwriting the previous value.

Syntax: <variable name>=<integer value>

Integer is used in the program to input and output the different values. Integers are mostly used to input the quantities given by the user and to output the values to the user.

The collection data type used in the program is:

A) Lists

Lists are the collection data type in Python which hold the comma separated values, enclosed within Squared Brackets. The items within the squared brackets do not need to be of the same data type. Each element of a list is assigned to a specific index number.

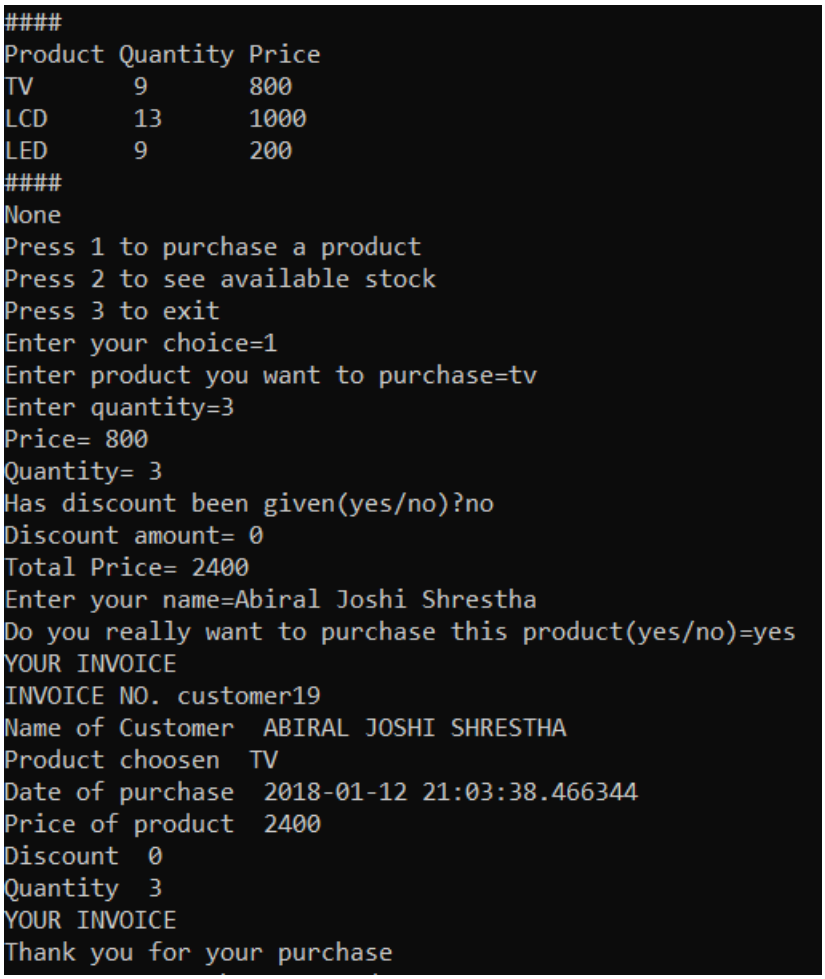
Syntax: <List name>= [<element name>, <element name>...]

List is used in the program to store the collective data types in the program. The main menu is created using lists.

9. Testing

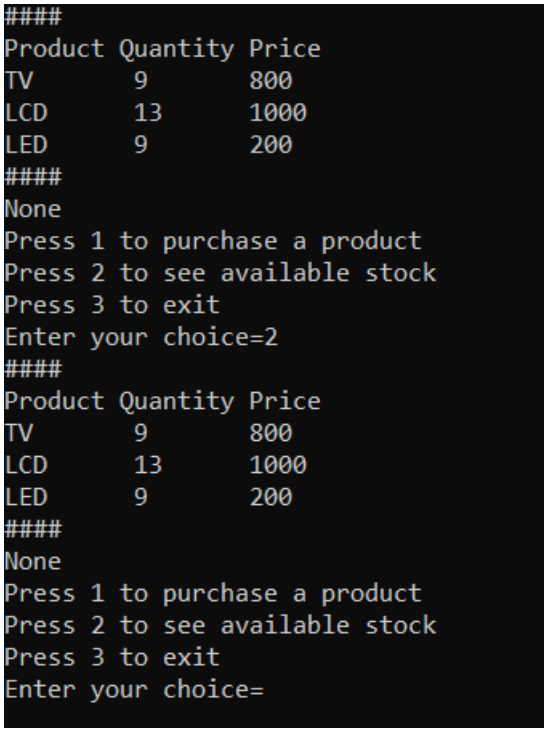
The program written by me was accurate and successful. To prove and verify this, the following tests were conducted.

A) Test 1



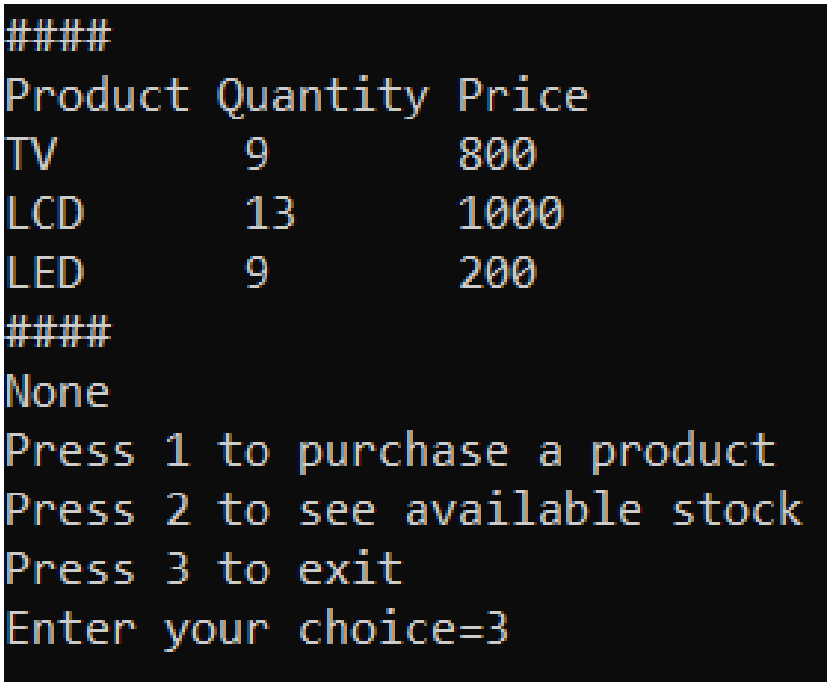
This is the first test. In this test, U can see from the above diagram a customer has brought a TV, no discount has been given, total price is displayed, user's name is asked and the Invoice is generated for the customer.

B) Test 2



This is the second test. In the above diagram, the customer wishes to check the available stock and it has been accordingly displayed.

C) Test 3



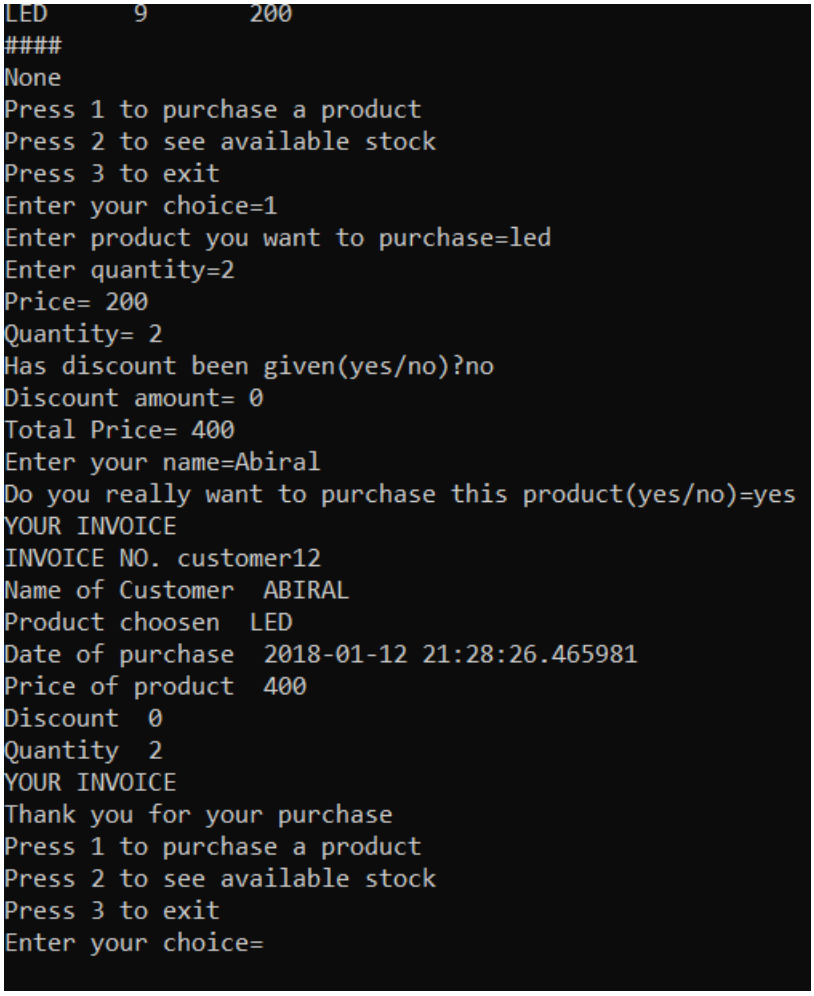
This is the third test. As soon as 3 is given as an input the program exits.

D)Test 4

```
####
Product Quantity Price
TV       9       800
LCD      13      1000
LED      9       200
####
None
Press 1 to purchase a product
Press 2 to see available stock
Press 3 to exit
Enter your choice=1
Enter product you want to purchase=2
-----------------------------------------------
NOT IN STOCK OR USER ENTERED WRONG PRODUCT NAME
-----------------------------------------------
Press 1 to purchase a product
Press 2 to see available stock
Press 3 to exit
Enter your choice=
```

This is the fourth test. It shows and verifies that the program ask the user to input a proper and valid data if invalid input is given.

E) Test 5



This is the fifth test. From the above diagram, you can see that the program can execute even when there are multiple inputs.

10. Research

Doing this coursework was a very hard task, but with the help of our module leader's guidance and with constant effort and research, made this task somewhat easier. Limitless resources for guidelines and reference ultimately made it possible to achieve all the aims of the coursework. Research has played a vital role in the successful completion of this project and the efficiency of the report.

A lot of web sites, journals and books were used as references in order to achieve the aims of this report.

Some of the websites used as reference:

A) https://www.datacamp.com/community/tutorials/data-structures-python (Anon., 2008)

This web site provides valuable information regarding the different types of data structures that can be used in python. The information is provided in a simple and transparent manner.

Some portions of this report were made much easier after referring and studying this web site. It provided knowledge about both Primitive and Non-Primitive data structures of Python programming language.

B) http://www.pythonforbeginners.com/loops/for-while-and-nested-loops-in-python (Anon., 2013)

This web site provides information about loops and more specifically about nested loops. It gave me a clearer idea about how to utilize nested loops.

This website was very useful to me as most parts of the program written by me use nested loops. I created the nested loops by referring to this web site.

C) https://www.programiz.com/python-programming/class (Bhatta, 2016)

This web site provides information regarding the use of Classes and functions in Python. This web sites provides simple as well as complex examples regarding the use of Classes and Functions.

Referring to this web site, it made it a lot easier for me to efficiently write my program and to divide the program into modules. Hence, the transparency and efficiency of the program improve drastically.

D) https://www.techopedia.com/definition/26859/information-management-system (Anon., 2013)

This web site provides detailed information about management system. It also focuses lightly on the forms of information management systems.

Being a Non-Management stream student throughout my schooling years, I had very little concept about information management system. So, in order to broaden my knowledge about information management systems I referred to this web site.

E) http://ecomputernotes.com/mis/what-is-mis/discuss-the-prerequisites-of-an-effective-mis (Anon., 2017)

This web site also provides information regarding information management systems. It focuses heavily on making an efficient system.

So as to improve the efficiency, usability and longevity of the system which I have created, I referred to this web site to learn about what makes an Information Management System efficient.

Some Journals used as references:

A) International Journal of Information Management

By: Philip Hills (Hills, 2015)

This journal provides knowledge about Information Management Systems in detail. It aims to bring its readers the very best analysis and discussion in the developing of information management.

Consulting this journal helped me getting to known about information management even better. It explained the positive and negative aspects of the various techniques that can be used in developing an information management system. Information from this journal made me aware about the requirements of the system to certain types of users.

B) Python for loop

By: Imtiaz Abedin (Abedin, 1998)

This journal provides information about loops. It primarily focuses nested loops and provides very complex, yet understandable, examples of long nested loops.

Consulting this journal helped me to efficiently use nested loops in Python programming. I have also incorporated some of the techniques used in this examples into my own program.

C) Python Programming

By: Jacek Artymiak (Artymiak, 2013)

This journal provides information about Python programming language and its uses. It also has examples of the different functions that can be used in Python.

Consulting this journal helped me resolve some of the doubts that I had while I was writing my program. I used this journal as a reference point if I faced any confusion. I also used many functions which I had learnt about in this journal.

D) Problem Solving with Algorithms and Data Structures using Python

By: Brad Miller and David Ranum (Ranum, 2000)

This journal provides information about the types of data structures and their uses in Python programming language. It contained simple algorithms on how to write complex Python programs.

Consulting this journal helped me make my program much more simple and efficient through the use of Python programming language. I implemented some of these techniques in my own program.

E) Retail Management Information System

By: Tara Duggan (Duggan, 2014)

This journal provides an idea about the use of an information management system in a retail store. It has detailed information regarding the needs of a retail store in the system.

Consulting this journal helped me better understand the type of Information Management System that I needed to create according to the need and requirement of the Electronics Store.

Some Books used as reference:

A) Data Structures and Algorithms with Python BY LEE, KENT D., (D., 1998)

This book explains the concepts and techniques required to write programs that can handle large amounts of data efficiency. This book presents a number of important algorithms that bring meaning to the prblems faced by computer programmers.

Consulting this book helped me in writing my own algorithm and use suitable techniques in writing my program to make it simple, yet, effective.

B) Essentials of Management Information Systems BY JANE LAUDON AND KENNETH C. LAUDON (Laudon, 2016)

This book explains about the essentials on an information management system. It complies all the necessary and basic requirements in the creation of an information management system.

Consulting this book helped me to better understand the concept of information management and how I could make my system even more efficient.

C) A Python Book BY DAVE KUHLMAN (Kuhlman, 2006)

This book contains information regarding the basic use of Python Programming. It focuses on the use of various functions in Python.

Consulting this book, I was able to remove any doubts and confusions I had while writing my program. I learnt in depth about the uses of different variables and various functions. It helped me to improve my coding efficiency.

D) Python Visual QuickStart Guide BY TOBY DONALDSON (Donalson, 2008)

This book has information about nested loops with very simple and basic examples. It made my concept about loop index variables clearer while using a nested loop.

Consulting this book helped me in creating the nested loops which I have used in my own program. Additionally, referring to this book helped me debug some portions of my own program.

E) Retail Management: A Strategic Approach BY BARRY R. BERMAN (Berman, 2014)

This book provides very detailed and technical scenarios of a retail store. It provides a strategic, decision-making approach that illustrates how retailers plan for, and adapt to today's changing and complex retail environment.

Consulting this book made me understand the needs of the user of the system better. It taught me that the program should be easily flexible and adaptable so as to improve the reliability and longevity of the system.

11) Conclusion and Formatting

All the tasks assigned in the coursework were finally completed. Many techniques were used and many issues were faced. But, with the help of my module leader I was able to remove those errors. Still, doing the tasks in this coursework were not easy and required lots of hard work and research. To make the coursework easier, it was divided into smaller modules and some objectives were set for each module. The program was written based on the algorithm and the flowchart. Finally, the program, which I wrote was tested in order to ensure that it had no bugs or errors, and that it could delivered accurate output. After the completion of all the assigned tasks, the submission was done before the deadline.

Although the main aim was to complete the coursework and submit it on time, it didn't have just a singular limited purpose. It also helped in developing various skills and taught many things about Python programming language and its practicable use which can even be useful in the future as a programmer. While doing this project, I received a better knowledge about Python as a programming language, its various in-built functions, and the use of variables, comments, nested loops, conditional statements and modularisation. At the end of the project, I feel much more comfortable using Python, using algorithms, using flowcharts and using pseudo codes. This kinds of projects are challenging and will indeed help in maing us as a good programmer.

Although the main aim of this project was intended for successful completion of all the given tasks, it was limited to just that. It can be very helpful to all the people who have curiosity about Python programming and the use of Information Management System. I won't that my program has the best coding for this type of tasks, the coding has still got lots of room for improvement. I tried my best to make this report as efficient as possible.

Lastly, I would like to thank everyone, my module leader and my fellow class mates, for helping me remove my confusions during the duration of this coursework. Also, to Islington College's librarians for providing me the books I used for reference to complete this coursework.

12) Bibliography

Bibliography

Abedin, I. (1998) Python for Loop. *Abedin*.

Anon. (2008) *Data Camp* [Online]. Available from: https://www.datacamp.com/community/tutorials/data-structures-python [Accessed 24 December 2017].

Anon. (2013) *Python For Beginners* [Online]. Available from: http://www.pythonforbeginners.com/loops/for-while-and-nested-loops-in-python [Accessed 24 December 2017].

Anon. (2013) *Techopedia* [Online]. Available from: https://www.techopedia.com/definition/26859/information-management-system [Accessed 3 January 2018].

Anon. (2017) *E-Computer Notes* [Online]. Available from: http://ecomputernotes.com/mis/what-is-mis/discuss-the-prerequisites-of-an-effective-mis [Accessed 14 December 2017].

Anon. (207) *E-Computer Notes* [Online]. Available from: http://ecomputernotes.com/mis/what-is-mis/discuss-the-prerequisites-of-an-effective-mis [Accessed 14 December 2017].

Artymiak, J. (2013) Python Programming. *Programming* .

Berman, B.R. (2014) *Retail Management: A Strategic Approach*.

Bhatta, R. (2016) *Programiz* [Online]. Available from: https://www.programiz.com/python-programming/class [Accessed 24 December 2017].

D., K. (1998) *Data Structures and Algorithms with Python*.

Donalson, T. (2008) *Python Visual QuickStart Guide*.

Duggan, T. (2014) Retail Management Information System. *Information Systems*, 2(23).

Hills, P. (2015) International Journal of Information Management. *Information Management Systems*.

Kuhlman, D. (2006) *A Python Book*.

Laudon, J. (2016) *Essentials of Management Information Systems*.

Ranum, B.M.&.D. (2000) Problem Solving with Algorithms and Data Structures using Python. *Python*.