BSc (Hons) Artificial Intelligence and Data Science

Module: CM1601 Programming Fundamentals

Coursework 1 Report

Module Leader: Ms. Sachinthani Perera

RGU Student ID : 2313082

IIT Student ID : 20220578

Student Name : B.G.C.Gomes

Abstract

This project is about an internet cafe, the things that happen there, and the things that are in the cafe. This system is specifically targeted at students who use internet cafes to do projects, assignments, and for enjoyment but do not have access to personal computers.

In this project, we must create an inventory system that allows users to add, delete, update, and view items. Also, as part of our project, we must define six dealers and interact with them as the user directs.

Six wholesale dealers, each of which sells high-quality items, have already provided the creator of this system with their contact information. Owing to the supplier's similarities, the creator has chosen to pick four dealers at random to buy the remaining items. John needs a way to control inventories because the cafe needs a lot of equipment.

Contents

Problem

Python command line applications should be used to develop this program. As described in the abstract, the project requires us to develop an inventory system that will benefit students without access to computers or the internet. For that, a system to manage the system's components, such as the number of computers, displays, keyboards, and so on, must be created. creator has received details of 6 dealers for buying items to the café. Creator is going to select four of those six dealers randomly. So we have to create a system to adding items, deleting items, updating items, viewing items and saving item details to a text file. Another task of this project is selecting four random dealers from the dealers. After selecting four dealers randomly we will be able to display the details of those four random dealers and system should display the particular item details when particular dealer name given.

Problem Understanding

You must construct these functions as you develop this project.

You must initially display the inventory system's menu. For the process to continue, the system needs input.

When a user entered the "AID" option, the system instructs them to add a new item to their inventory. The system needs to obtain the following data before adding an item. such includes the item's code, name, brand, price, quantity, category, and date of purchase. The system can successfully add an item to the inventory by receiving this information.

A "DID" input should be received by the system when deleting an item. The owner should be able to erase item details using the system's item code search function.

System must receive "VID" as input in order to view an item. System should output the current total of the items that have previously been purchased and show item details in descending order while taking item code into account. All the information in this table needs to be correctly formatted.

The user must select "SID" as the input option if they wish to save these details to a text file. The owner should always have the option to save the item's details to the same text file using the system.

When "SDD" is selected, the system should simulate a drawing at random and choose four dealers from a text file in accordance. When the selection is complete, the message "4 Dealers are Picked Randomly" is displayed. The following information would be known to a dealer. Name, phone number, address, and products of the dealers (3 items for each dealer). Name, Brand, Price, and Quantity are necessary item data.

The input must be "VRL" if the user wishes to see all the information about the chosen dealers. This will cause the system to display all of the dealer information for the randomly chosen dealer. sorted according to the location.

When "LDI" is entered into the system, it ought to be able to display the products when one of the randomly chosen dealer names is entered.

This is how the project to control inventories works.

Python Code (Source Code)

Main.py

```
from tabulate import tabulate
import json
import random
import inventory_dictionary
items_dictionary = {}

def inventory_introduction():
    from tabulate import tabulate

    head=[["WELCOME TO ONE NET CAFE - INVENTORY SYSTEM" ] ]

    print(tabulate(head,tablefmt="double_grid"))

    #table = tabulate(data, tablefmt="grid")

def main():
    #from colorama import Fore
    print( """\033[3m
           • Type AID for adding item details.
           • Type DID for deleting item details.
           • Type UID for updating item details.
           • Type VID for viewing the items table.
```

```
          • Type SID for saving the item details to the text file at any time.
          • Type SDD for selecting four dealers randomly from a file.
          • Type VRL for displaying all the details of the randomly selected
dealers.
          • Type LDI for display the items of the given dealer.
          • Type ESC to exit the program.
          \033[0m""")

def inventory_load():
    global items_dictionary
    try:
        with open("inventory.txt", "r") as file:
            items_dictionary = json.load(file)
    except FileNotFoundError:
        pass

def inventory_save():
    with open("inventory.txt", "w") as f:
        for i_code, i_details in items_dictionary.items():
            item_record = f" 'Item code'= {i_code}, 'Item name'={i_details['Item
Name']},'Item brand'={i_details['Item Brand']}, 'Item price'={i_details['Item
Price']},'Quantity'={i_details['Item Quantity']}, 'category'= {i_details['Item
Category']} 'Purchased Date'= {i_details['Purchased Date']}\n"
            f.write(item_record)

def adding_item_details():
    i_code = ""
    while not i_code:
        i_code = input("Enter Item Code: ")
        i_code=i_code.strip()
        if i_code in items_dictionary:
            print(f"\033[1m\033[94m{i_code} item already in inventory.\033[0m")
            i_code = ""
            continue

    i_name = ""
    while not i_name:
        i_name = input("Enter Item Name: ")
        i_name=i_name.strip()

    i_brand = ""
    while not i_brand:
        i_brand = input("Enter Item Brand: ")
        i_brand=i_brand.strip()

    i_price = ""
    while not i_price:
        try:
            i_price = float(input("Enter Item Price: "))
            #i_price=i_price.strip()
        except ValueError:
            print(f"\033[1m\033[31mInvalid input. Please enter a valid number
value.\033[0m")
            continue

    i_quantity = ""
    while not i_quantity:
        try:
            i_quantity = int(input("Enter Item Quantity: "))
        except ValueError:
            print(f"\033[1m\033[31mInvalid input. Please enter a valid integer
value.\033[0m")
            continue

    i_category = ""
    while not i_category:
        i_category = input("Enter Item Category: ")
        i_category=i_category.strip()
```

```
    item_purchased_date = ""
    while not item_purchased_date:
        item_purchased_date = input("Enter Purchased Date (DD/MM/YYYY): ")
        item_purchased_date=item_purchased_date.strip()

        i_details = {"I Name": i_name,
                     "I Brand": i_brand,
                     "I Price": i_price,
                     "I Quantity": i_quantity,
                     "I Category": i_category,
                     "I Purchased Date": item_purchased_date}

        items_dictionary[i_code] = i_details
        print(f"\033[1m\033[92m{i_name} ({i_code}) added to inventory.\033[0m")
        inventory_save()

def deleting_item():
    i_code = input("Enter Item Code: ")
    if i_code in items_dictionary:
        del items_dictionary[i_code]
        print(f"\033[1m\033[93m{i_code} deleted from inventory.\033[0m")
        inventory_save()
    else:
        print(f"\033[1m\033[31m{i_code} not found in inventory.\033[0m")

def updating_item():
    i_code = input("Enter Item Code: ")
    if i_code in items_dictionary:
        print(f"\033[1m\033[36mNew details for the item (Dont Enter Anything If you
need to keep the recent record.):\033[0m")
        i_name = input(f"Previous Item Name: {items_dictionary[i_code]['I
Name']}\nNew Item Name: ")
        i_name=i_name.strip()
        i_brand = input(f"Previous Item Brand: {items_dictionary[i_code]['I
Brand']}\nNew Item Brand: ")
        i_brand=i_brand.strip()
        while True:
            try:
                i_price = float(input(f"Previous Item Price:
{items_dictionary[i_code]['I Price']}\nNew Item Price: "))
                break
            except ValueError:
                print(f"\033[1m\033[31mInvalid input. Please enter a valid number
value.\033[0m")
        while True:
            try:
                i_quantity = int(input(f"Previous Item Quantity:
{items_dictionary[i_code]['I Quantity']}\nNew Item Quantity: "))
                break
            except ValueError:
                print(f"\033[1m\033[31mInvalid input. Please enter a valid number
value.\033[0m")
        i_category = input(f"Previous Item Category: {items_dictionary[i_code]['I
Category']}\nNew Item Category: ")
        i_category=i_category.strip()
        item_purchased_date = input(f"Previous Purchased Date:
{items_dictionary[i_code]['I Purchased Date']}\nNew Purchased Date (DD/MM/YYYY): ")
        item_purchased_date=item_purchased_date.strip()

        if i_name:
            items_dictionary[i_code]['I Name'] = i_name
        if i_brand:
            items_dictionary[i_code]['I Brand'] = i_brand
        if i_price:
            items_dictionary[i_code]['I Price'] = float(i_price)
        if i_quantity:
            items_dictionary[i_code]['I Quantity'] = int(i_quantity)
```

```
        if i_category:
            items_dictionary[i_code]['I Category'] = i_category
        if item_purchased_date:
            items_dictionary[i_code]['I Purchased Date'] = item_purchased_date
        inventory_save()
        print(f"\033[1m\033[92m {i_code} ({i_name}) details updated.\033[0m")
        #print(f"{i_code}{i_name} details updated.")
    else:
        print(f"\033[1m\033[31m{i_code} not found in inventory.\033[0m") #red

def viewing_items():
    if not items_dictionary:
       # print("No items found!")
        print("\033[1m\033[31mNo items found!\033[0m")
    else:
        items = []
        total_purchased_items = 0
        for i_code, item_details in items_dictionary.items():
            items.append([i_code, item_details["I Name"], item_details["I Brand"],
item_details["I Price"], item_details["I Quantity"], item_details["I Category"],
item_details["I Purchased Date"]])
            total_purchased_items += item_details["Item Quantity"]

        items = sorted(items, key=lambda x: x[0], reverse=True)

        headers = ["\033[1mItem code\033[0m", "\033[1mItem Name\033[0m",
"\033[1mItem Brand\033[0m","\033[1mItem Price\033[0m", "\033[1mItem
Quantity\033[0m", "\033[1mItem Category\033[0m","\033[1mPurchased Date\033[0m"]
        print(tabulate(items, headers=headers, tablefmt="heavy_grid"))

        print("\033[1m\033[93mTotal Purchased Items:
{}\033[0m".format(total_purchased_items))

while True:

    inventory_introduction()
    main()

    choice = input("Enter your choice: ")
    choice = choice.upper()
    if choice == 'AID':
        adding_item_details()
        print()

    elif choice == 'DID':
        deleting_item()
    elif choice == 'VID':
        viewing_items()
        print()
    elif choice == 'UID':
        updating_item()
        print()


    elif choice == 'SDD':
        inventory_dictionary.ddd()
        try:

            with open('dealers.txt', 'r') as file:
                dealers_json = file.read()
                dealers = json.loads(dealers_json)

            random_dealers = random.sample(list(dealers.keys()), 4)
            print("\033[1m\033[34m4 Dealers are selected Randomly\033[0m")
            print()
        except:
            print("\033[1m\033[31mFile Not Found!\033[0m")
```

```
    elif choice == 'VRL':

        try:
            for i in range(len(random_dealers)):
                for j in range(len(random_dealers) - i - 1):
                    if dealers[random_dealers[j]]['Dealer_Location'] >
dealers[random_dealers[j + 1]]['Dealer_Location']:
                        random_dealers[j], random_dealers[j + 1] = random_dealers[j
+ 1], random_dealers[j]

            rows = []
            for dealer in random_dealers:
                d_row = [dealer, dealers[dealer]['Telephone_Number'],
dealers[dealer]['Dealer_Location']]
                rows.append(d_row)
                for item in dealers[dealer]['items']:
                    i_row = [None, None, None, item['name'], item['brand'],
item['price'], item['quantity']]
                    rows.append(i_row)

            headers = ['Dealer Name', 'Contact Number', 'Location', 'Item Name',
'Brand','price', 'Quantity']
            print(tabulate(rows, headers=headers))
        except:
            print("\033[1m\033[91mYou have not selected any dealers yet!!!\033[0m")

    elif choice == 'LDI':
        try:
            dealer_name = input("Enter Dealer Name ( Please select from the
randomly selected dealer table ) : ")
            dealer_name=dealer_name.strip()

            if dealer_name in random_dealers:
                print(f"\033[1m\033[36mDealer name:  {dealer_name}\033[0m ")


                get_dealer=[]

                for item in dealers[dealer_name]['items']:
                    i_row = [item['name'], item['brand'], item['quantity'],
item['price']]
                    get_dealer.append(i_row)
                headers = ['\033[1mItem Name\033[0m', '\033[1mBrand\033[0m',
'\033[1mQuantity\033[0m','\033[1mPrice\033[0m']
                print(tabulate(get_dealer, headers=headers, tablefmt='heavy_grid'))

            else:
                #print("Dealer not found!")
                print("\033[1m\033[31mDealer is not in the system (Please use
randomly selected dealers...) \033[0m")
        except:
            print("\033[1m\033[31mThere are no any randomly selected dealers.
Please try again the 'SDD'...\033[0m")

    elif choice=='SID':
        inventory_save()
        print(f"\033[1m\033[92mInventory saved.\033[0m")

    elif choice == 'ESC':
        print("Thankyou")
        break

    else:
        print("\033[1m\033[31mInvalid choice.Try again.\033[0m")
```

Inventory_dictionary.py

```
import json
dealers = {
    'Gawesh Gomes': {
        'Telephone_Number': '+94761328236',
        'Dealer_Location':'Colombo',
        'items': [
            {
                'name': 'dell XPS Laptop',
                'brand': 'DELL',
                'price': 'RS. 590000.00',
                'quantity': '50'
            },
            {
                'name': 'Gaming keyboard',
                'brand': 'asus',
                'price': 'RS. 4900.00',
                'quantity': '30'
            },
            {
                'name': 'CPU',
                'brand': 'DELL',
                'price': 'RS. 220000.00',
                'quantity': '20'
            }
        ]
    },
    'Radil Damsara': {
        'Telephone_Number': '+94714325437',
        'Dealer_Location':'Kaluthara',
        'items': [
            {
                'name': 'flash drive',
                'brand': 'HP',
                'price': 'Rs. 12000.00',
                'quantity': '40'
            },
            {
                'name': 'Dell XPS 13',
                'brand': 'Dell',
                'price': 'Rs. 550000.00',
                'quantity': '15'
            },
            {
                'name': 'heat fan pro',
                'brand': 'Dell',
                'price': 'Rs. 45000.00',
                'quantity': '5 '
            }
        ]
    },
    'Devindi Perera': {
        'Telephone_Number': '+94754567242',
        'Dealer_Location':'Kiribathgoda',
        'items': [
            {
                'name': 'Redmi note 9 pro',
                'brand': 'Redmi',
                'price': 'Rs. 70000.00',
                'quantity': '8'
            },
            {
```

```
                'name': 'Gaming mouse',
                'brand': 'Asus',
                'price': 'Rs. 3900',
                'quantity': '16'
            },
            {
                'name': 'monitor',
                'brand': 'HP',
                'price': 'Rs. 80000.00',
                'quantity': '12'
            }
        ]
    },
    'Diyathma_wijewardhana': {
        'Telephone_Number': '+94769765434',
        'Dealer_Location':'Moratuwa',
        'items': [
            {
                'name': 'Iphone 14 pro max',
                'brand': 'Apple',
                'price': 'Rs. 649000.00',
                'quantity': '10'
            },
            {
                'name': 'computer lamp',
                'brand': 'orange',
                'price': 'Rs. 4800.00',
                'quantity': '50'
            },
            {
                'name': 'Iphone charger',
                'brand': 'apple',
                'price': 'Rs. 12000.00',
                'quantity': '40'
            }
        ]
    },
    'Sehandu_Siriwardhana': {
        'Telephone_Number': '+94786756453',
        'Dealer_Location':'kurunagala',
        'items': [
            {
                'name': 'laptop bags',
                'brand': 'asus',
                'price': 'Rs. 2500.00',
                'quantity': '50'
            },
            {
                'name': 'Headphone',
                'brand': 'samsung',
                'price': 'Rs. 3000.00',
                'quantity': '24'
            },
            {
                'name': 'SSD',
                'brand': 'Sandisk',
                'price': 'Rs. 10000.00',
                'quantity': '35'
            }
        ]
    },
    'Malindu_Dilshan': {
        'Telephone_Number': '+94772534657',
        'Dealer_Location':'Panadura',
        'items': [
            {
                'name': 'Memory card',
                'brand': 'Sandisk',
```

```
                'price': 'Rs. 4000.00',
                'quantity': '20'
            },
            {
                'name': 'Memory card readers',
                'brand': 'Ugreen',
                'price': 'Rs. 1000.00',
                'quantity': '45'
            },
            {
                'name': 'projector',
                'brand': 'ViewSonic',
                'price': 'Rs. 120000.00',
                'quantity': '18'
            }
        ]
    }
}


# Convert dictionary to JSON string
def ddd():
    dealers_json = json.dumps(dealers)

    # Write JSON string to file
    with open('dealers.txt', 'w') as file:
        file.write(dealers_json)
```

Functions

Introduction()

```
def inventory_introduction():
    from tabulate import tabulate

    head=[["WELCOME TO ONE NET CAFE - INVENTORY SYSTEM" ] ]

    print(tabulate(head,tablefmt="double_grid"))

    #table = tabulate(data, tablefmt="grid")
```

After running this function system will print the welcome message inside a table cell.

Main()

```
def main():
    #from colorama import Fore
    print( """\033[3m
            • Type AID for adding item details.
            • Type DID for deleting item details.
            • Type UID for updating item details.
            • Type VID for viewing the items table.
            • Type SID for saving the item details to the text file at any time.
            • Type SDD for selecting four dealers randomly from a file.
            • Type VRL for displaying all the details of the randomly selected
dealers.
            • Type LDI for display the items of the given dealer.
            • Type ESC to exit the program.
           \033[0m""")
```

After running this main function system display the main menu of the system.

```
    • Type AID for adding item details.
    • Type DID for deleting item details.
    • Type UID for updating item details.
    • Type VID for viewing the items table.
    • Type SID for saving the item details to the text file at any time.
    • Type SDD for selecting four dealers randomly from a file.
    • Type VRL for displaying all the details of the randomly selected dealers.
    • Type LDI for display the items of the given dealer.
    • Type ESC to exit the program.
```

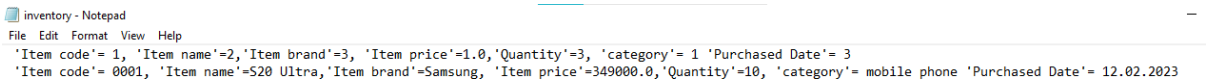
Load_inventory() and save_inventory()

```
def inventory_load():
    global items_dictionary
    try:
        with open("inventory.txt", "r") as file:
            items_dictionary = json.load(file)
    except FileNotFoundError:
        pass

def inventory_save():
    with open("inventory.txt", "w") as f:
```

```
        for i_code, i_details in items_dictionary.items():
            item_record = f" 'Item code'= {i_code}, 'Item name'={i_details['Item
Name']},'Item brand'={i_details['Item Brand']}, 'Item price'={i_details['Item
Price']},'Quantity'={i_details['Item Quantity']}, 'category'= {i_details['Item
Category']} 'Purchased Date'= {i_details['Purchased Date']}\n"
            f.write(item_record)
```

After running this function system can store their item details and save them in a text file.



Add item()

```
def adding_item_details():
    i_code = ""
    while not i_code:
        i_code = input("Enter Item Code: ")
        i_code=i_code.strip()
        if i_code in items_dictionary:
            print(f"\033[1m\033[94m{i_code} item already in inventory.\033[0m")
            i_code = ""
            continue

    i_name = ""
    while not i_name:
        i_name = input("Enter Item Name: ")
        i_name=i_name.strip()

    i_brand = ""
    while not i_brand:
        i_brand = input("Enter Item Brand: ")
        i_brand=i_brand.strip()

    i_price = ""
    while not i_price:
        try:
            i_price = float(input("Enter Item Price: "))
            #i_price=i_price.strip()
        except ValueError:
            print(f"\033[1m\033[31mInvalid input. Please enter a valid number
value.\033[0m")
            continue
```

```
    i_quantity = ""
    while not i_quantity:
        try:
            i_quantity = int(input("Enter Item Quantity: "))
        except ValueError:
            print(f"\033[1m\033[31mInvalid input. Please enter a valid integer
value.\033[0m")
            continue

    i_category = ""
    while not i_category:
        i_category = input("Enter Item Category: ")
        i_category=i_category.strip()

    item_purchased_date = ""
    while not item_purchased_date:
        item_purchased_date = input("Enter Purchased Date (DD/MM/YYYY): ")
        item_purchased_date=item_purchased_date.strip()

        i_details = {"I Name": i_name,
                     "I Brand": i_brand,
                     "I Price": i_price,
                     "I Quantity": i_quantity,
                     "I Category": i_category,
                     "I Purchased Date": item_purchased_date}

        items_dictionary[i_code] = i_details
        print(f"\033[1m\033[92m{i_name} ({i_code}) added to inventory.\033[0m")
        inventory_save()
```

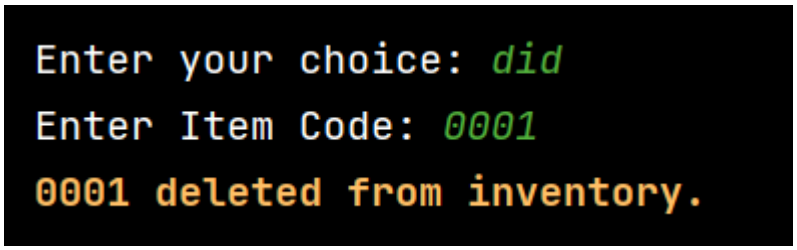
After running this function the system will be able to get details of an item.

```
Enter your choice: AID
Enter Item Code: 0001
Enter Item Name: S20 Ultra
Enter Item Brand: Samsung
Enter Item Price: 349000
Enter Item Quantity: 10
Enter Item Category: mobile phone
Enter Purchased Date (DD/MM/YYYY): 12.02.2023
S20 Ultra (0001) added to inventory.
```

Delete_item()

```
def deleting_item():
    i_code = input("Enter Item Code: ")
    if i_code in items_dictionary:
        del items_dictionary[i_code]
        print(f"\033[1m\033[93m{i_code} deleted from inventory.\033[0m")
        inventory_save()
    else:
        print(f"\033[1m\033[31m{i_code} not found in inventory.\033[0m")
```

Items can be removed from the inventory system using the delete function. The system ought to remove all the information about that specific item from the inventory system once a user requests a specific item code.



Update_item()

```
def updating_item():
    i_code = input("Enter Item Code: ")
    if i_code in items_dictionary:
        print(f"\033[1m\033[36mNew details for the item (Dont Enter Anything If you need to keep the recent record.):\033[0m")
        i_name = input(f"Previous Item Name: {items_dictionary[i_code]['I Name']}\nNew Item Name: ")
        i_name=i_name.strip()
        i_brand = input(f"Previous Item Brand: {items_dictionary[i_code]['I Brand']}\nNew Item Brand: ")
        i_brand=i_brand.strip()
        while True:
            try:
                i_price = float(input(f"Previous Item Price: {items_dictionary[i_code]['I Price']}\nNew Item Price: "))
                break
            except ValueError:
                print(f"\033[1m\033[31mInvalid input. Please enter a valid number value.\033[0m")
        while True:
            try:
                i_quantity = int(input(f"Previous Item Quantity: {items_dictionary[i_code]['I Quantity']}\nNew Item Quantity: "))
                break
            except ValueError:
                print(f"\033[1m\033[31mInvalid input. Please enter a valid number value.\033[0m")
        i_category = input(f"Previous Item Category: {items_dictionary[i_code]['I Category']}\nNew Item Category: ")
        i_category=i_category.strip()
        item_purchased_date = input(f"Previous Purchased Date: {items_dictionary[i_code]['I Purchased Date']}\nNew Purchased Date (DD/MM/YYYY): ")
```

```
        item_purchased_date=item_purchased_date.strip()

        if i_name:
            items_dictionary[i_code]['I Name'] = i_name
        if i_brand:
            items_dictionary[i_code]['I Brand'] = i_brand
        if i_price:
            items_dictionary[i_code]['I Price'] = float(i_price)
        if i_quantity:
            items_dictionary[i_code]['I Quantity'] = int(i_quantity)
        if i_category:
            items_dictionary[i_code]['I Category'] = i_category
        if item_purchased_date:
            items_dictionary[i_code]['I Purchased Date'] = item_purchased_date
        inventory_save()
        print(f"\033[1m\033[92m {i_code} ({i_name}) details updated.\033[0m")
        #print(f"{i_code}{i_name} details updated.")
    else:
        print(f"\033[1m\033[31m{i_code} not found in inventory.\033[0m") #red
```

After running this function we can update every information of an item.

```
Enter your choice: uid
Enter Item Code: 0002
New details for the item (Dont Enter Anything If you need to keep the recent record.):
Previous Item Name: computer
New Item Name: Laptop computer
Previous Item Brand: dell
New Item Brand: Asus
Previous Item Price: 460000.0
New Item Price: 590000
Previous Item Quantity: 15
New Item Quantity: 20
Previous Item Category: laptop
New Item Category: laptop
Previous Purchased Date: 15.03.2023
New Purchased Date (DD/MM/YYYY): 30.11.2023
 0002 (Laptop computer) details updated.
```

View_item()

```
def viewing_items():
    if not items_dictionary:
       # print("No items found!")
        print("\033[1m\033[31mNo items found!\033[0m")
    else:
        items = []
        total_purchased_items = 0
        for i_code, item_details in items_dictionary.items():
            items.append([i_code, item_details["I Name"], item_details["I Brand"],
item_details["I Price"], item_details["I Quantity"], item_details["I Category"],
item_details["I Purchased Date"]])
            total_purchased_items += item_details["Item Quantity"]
        #headers = ["Item ID", "Item Name", "Item Brand", "Item Price", "Item
Quantity", "Item Category", "Purchased Date"]
        #print(tabulate(items, headers=headers, tablefmt="heavy_grid"))
        items = sorted(items, key=lambda x: x[0], reverse=True)

        headers = ["\033[1mItem code\033[0m", "\033[1mItem Name\033[0m",
"\033[1mItem Brand\033[0m","\033[1mItem Price\033[0m", "\033[1mItem
Quantity\033[0m", "\033[1mItem Category\033[0m","\033[1mPurchased Date\033[0m"]
```

```
        print(tabulate(items, headers=headers, tablefmt="heavy_grid"))
        #print("Total Purchased Items: {}".format(total_purchased_items))
        print("\033[1m\033[93mTotal Purchased Items:
{}\033[0m".format(total_purchased_items))
```

By entering "VID" and the total number of the item, the user can display the inventory's contents in decreasing order.

```
Enter your choice: vid
```

Item code	Item Name	Item Brand	Item Price	Item Quantity	Item Category	Purchase
0002	Viva book	Asus	420000	18	laptop	18.12.20
0001	S20 Ultra	Samsung	349000	15	Mobile Phone	12.11.20

```
Total Purchased Items: 33
```

Test Cases

Test case	Input	Expected Output	Actual Output	Remark
01	Enter 'AID' for adding items Ex: Enter your choice : AID	Output → Enter Item Code: Enter Item Name: Enter Item Brand: Enter Item Price: Enter Item Quantity: Enter Item Category: Enter Purchased Date (DD/MM/YYYY):	Output → Enter Item Code: Enter Item Name: Enter Item Brand: Enter Item Price: Enter Item Quantity: Enter Item Category: Enter Purchased Date (DD/MM/YYYY):	Test case pass
02	Try to skip the details	Until the necessary information is provided,	Until the necessary information is provided,	Test case pass

	without giving it.	the system will repeatedly make the same message.	the system will repeatedly make the same message.	
03	Give the same item code again and again.	Instead of displaying an error message, ask for a different item code to print a message to the user.	Instead of displaying an error message, ask for a different item code to print a message to the user.	Test case pass
04	Giving a string value for the item price	Print a message to the user to enter a valid integer number	Print a message to the user to enter a valid integer number	Test case pass
05	Giving a string value for item quantity	Print a message to the user to enter a valid integer number	Print a message to the user to enter a valid integer number	Test case pass
06	Enter 'DID' for deleting item	Print a message to the user the item is deleted. Display item deleted from inventory.	Print a massage to the user the item is deleted. Display item deleted from inventory.	Test case pass
07	Enter a wrong item code for 'DID'	Print a message to user "item not found in inventory"	Print a message to user "item not found in inventory"	Test case pass
08	Enter 'UID' for update items	Print a input statements such as, New details for the item (Don't Enter Anything If you need to keep the recent record.): Previous Item Name: Viva book New Item Name: Previous Item Brand: Asus New Item Brand: Previous Item Price: 420000.0 New Item Price:	Print a input statements such as, New details for the item (Don't Enter Anything If you need to keep the recent record.): Previous Item Name: Viva book New Item Name: Previous Item Brand: Asus New Item Brand: Previous Item Price: 420000.0 New Item Price:	Test case pass

		Previous Item Quantity: 18 New Item Quantity: Previous Item Category: laptop New Item Category: Previous Purchased Date: 18.12.2023 New Purchased Date (DD/MM/YYYY): 0002 (notebook) details updated.	Previous Item Quantity: 18 New Item Quantity: Previous Item Category: laptop New Item Category: Previous Purchased Date: 18.12.2023 New Purchased Date (DD/MM/YYYY): 0002 (notebook) details updated.	
09	Enter a wrong item code for 'UID'	Print a message to user "item not found in inventory"	Print a message to user "item not found in inventory"	Test case pass
10	Enter 'VID' for view item	Print a table contains all items and total purchased items in the inventory system	Print a table contains all items and total purchased items in the inventory system	Test case pass
11	Enter 'SID' for saving the item details to the text file at any time.	Print a message to user and save the inventory details in a text file. Ex: "Inventory saved."	Print a message to user and save the inventory details in a text file. Ex: "Inventory saved."	Test case pass
12	Enter 'SDD' for selecting four dealers randomly from a file.	Print a message to user Ex: " 4 Dealers are selected Randomly"	Print a message to user Ex: " 4 Dealers are selected Randomly"	Test case pass
13	Enter "VRL" for displaying all the details of the randomly selected dealers.	Print " 'dealer name' , 'Contact number' , 'Dealer location' , 'Item name' , 'Brand' , 'price' , 'quantity'" in a table	Print " 'dealer name' , 'Contact number' , 'Dealer location' , 'Item name' , 'Brand' , 'price' , 'quantity'" in a table	Test case pass
14	If User enters 'VRL' option without entering 'SDD' option	Print a message to user Ex: "You have not selected any dealers yet"	Print a message to user Ex: "You have not selected any dealers yet"	Test case pass
15	Enter 'LDI' for display the items of the given dealer.	Ask user to enter a dealer name for display the details about the dealer. Ex: "Enter Dealer Name (Please select from the randomly selected dealer table) :"	Ask user to enter a dealer name for display the details about the dealer. Ex: "Enter Dealer Name (Please select from the randomly selected dealer table) :"	Test case pass
16	Enter a wrong name what	Print a message to user	Print a message to user	Test case pass

	not in the system	Ex: “Dealer is not in the system (Please use randomly selected dealers...)”	Ex: “Dealer is not in the system (Please use randomly selected dealers...)”	
17	Enter ‘ESC’ to exit the program.	Print a massage to user the program was end. Ex: “Thankyou”	Print a massage to user the program was end. Ex: “Thankyou”	Test case pass

Test Case 01

Enter ‘AID’ for adding items

Ex: Enter your choice : AID

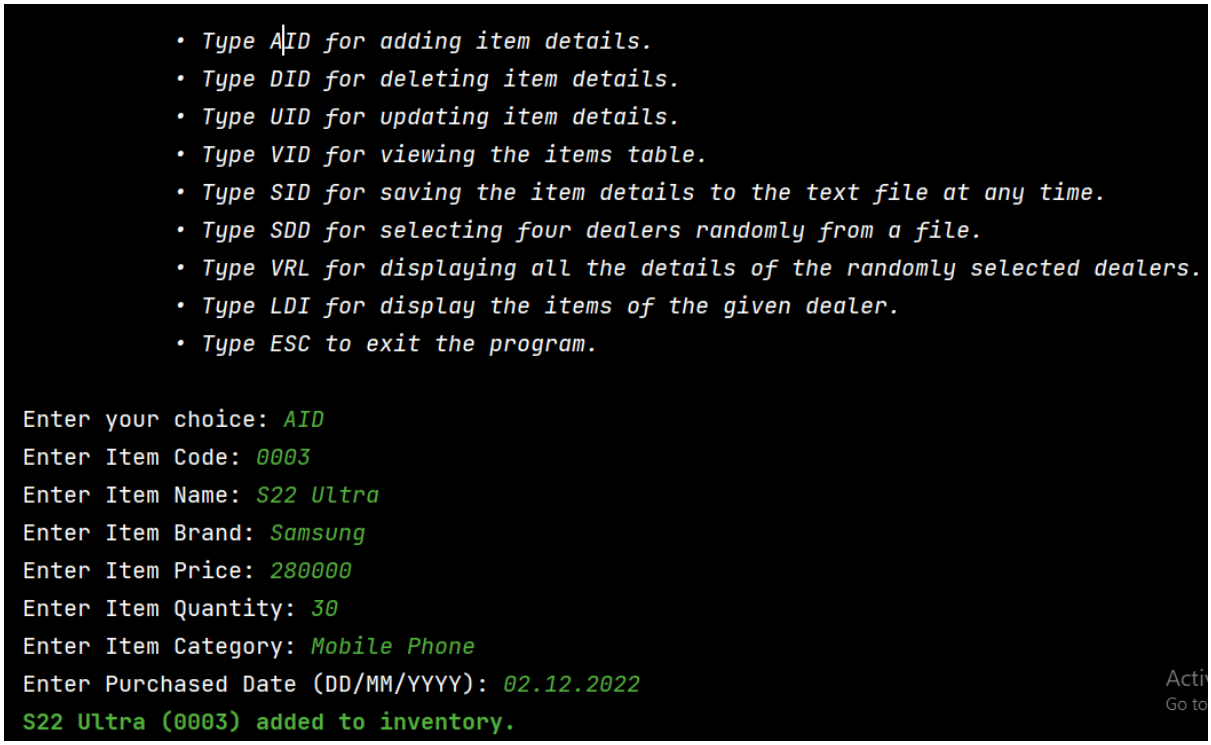


Figure 1

Test case 02

Try to skip the details without giving it. (exception handling)

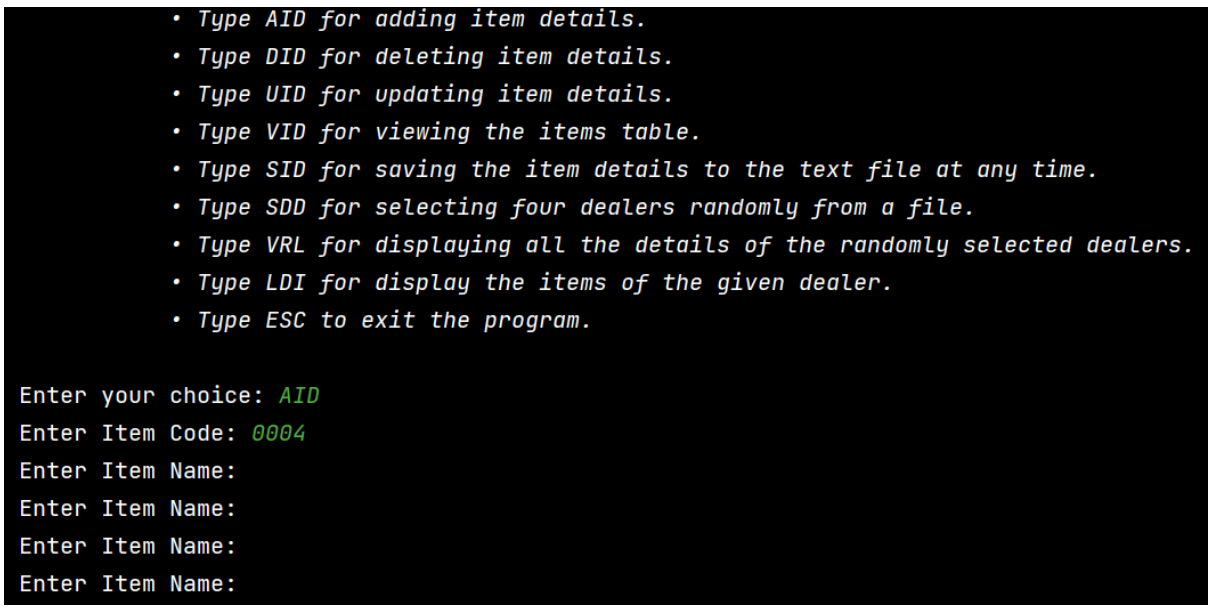


Figure 2

Test Case 03

Give the same item code again and again (exception handling)

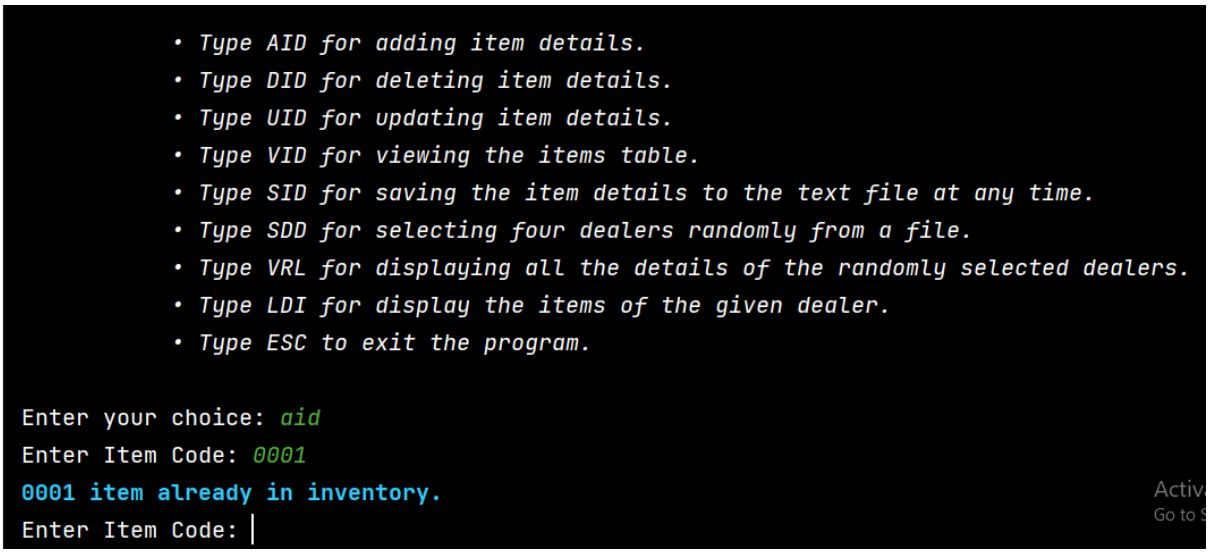
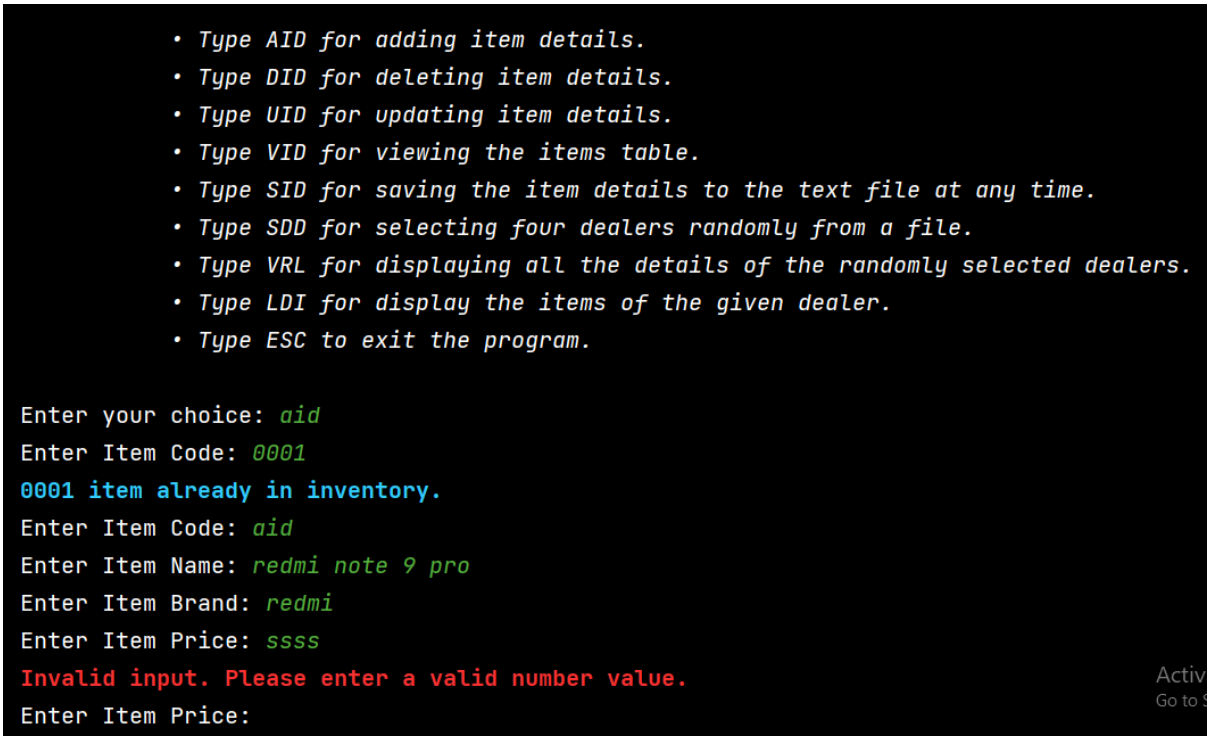


Figure 3

Test Case 04

Giving a string value for the item price. (Exception handling)



Test case 05

Giving a string value for item quantity (Exception handling)

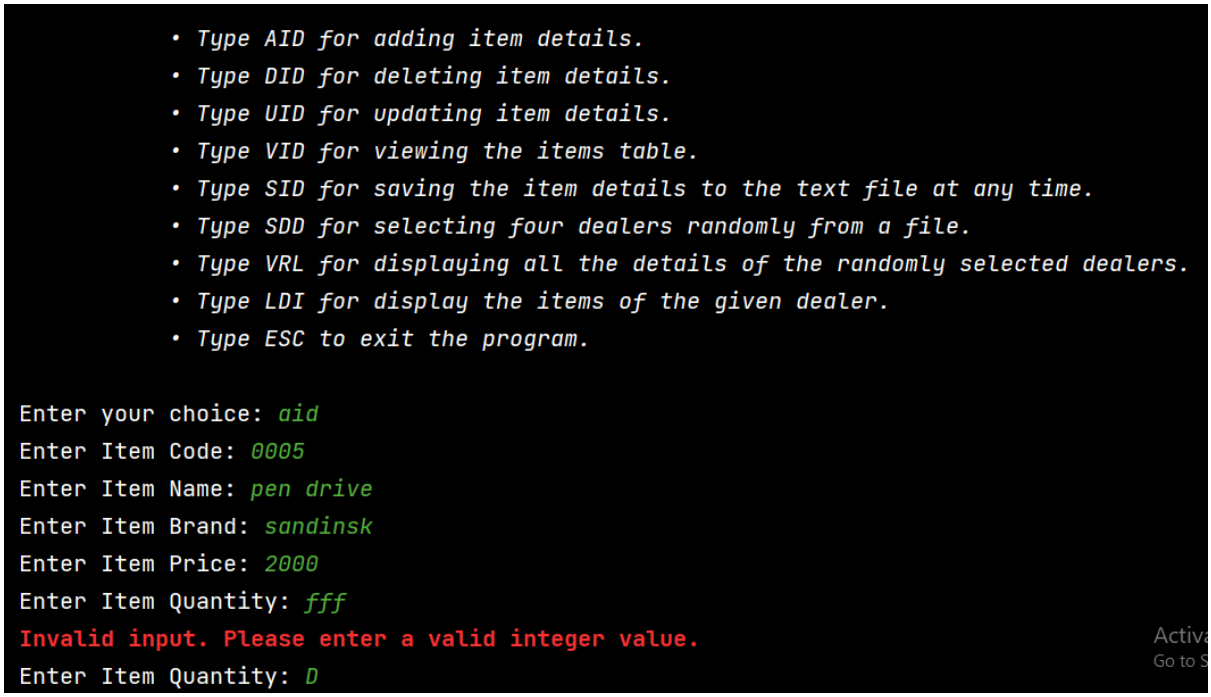


Figure 4

Test case 06

Enter ‘DID’ for deleting item

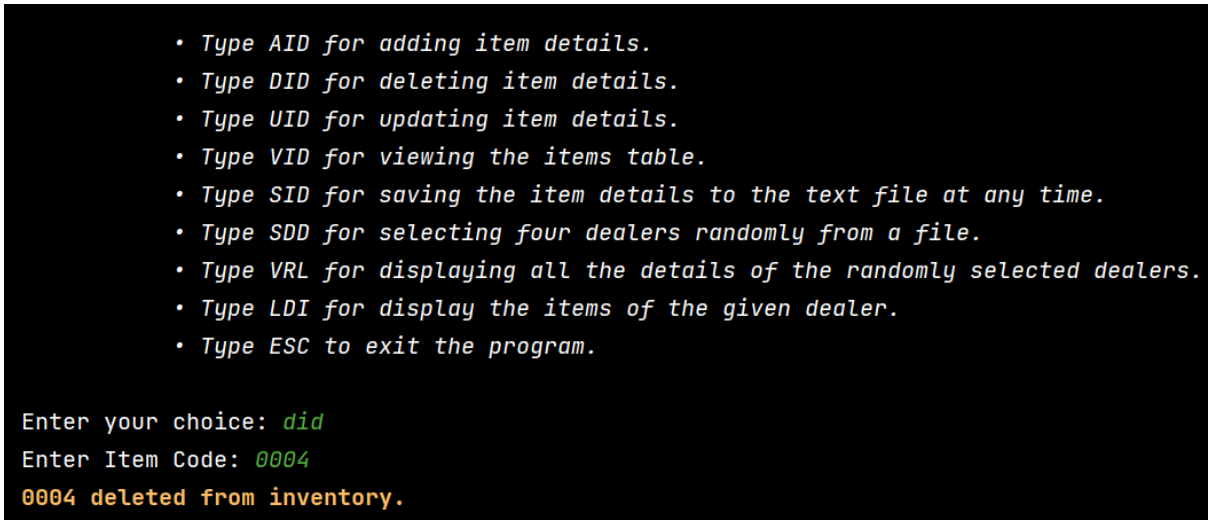


Figure 5

Test Case 07

Enter a wrong item code for ‘DID’ (exception handling)

```
            • Type AID for adding item details.
            • Type DID for deleting item details.
            • Type UID for updating item details.
            • Type VID for viewing the items table.
            • Type SID for saving the item details to the text file at any time.
            • Type SDD for selecting four dealers randomly from a file.
            • Type VRL for displaying all the details of the randomly selected dealers.
            • Type LDI for display the items of the given dealer.
            • Type ESC to exit the program.

Enter your choice: did
Enter Item Code: 0009
0009 not found in inventory.
```

Figure 6

Test case 08

Enter ‘UID’ for update items

```
Enter your choice: UID
Enter Item Code: 0002
New details for the item (Dont Enter Anything If you need to keep the recent record.):
Previous Item Name: I Phone 14 pro max
New Item Name: I Phone 12 pro max
Previous Item Brand: Apple
New Item Brand:
Previous Item Price: 550000.0
New Item Price: 360000
Previous Item Quantity: 35
New Item Quantity: 30
Previous Item Category: Mobile Phone
New Item Category:
Previous Purchased Date: 11.11.2022
New Purchased Date (DD/MM/YYYY): 12.09.2022
 0002 (I Phone 12 pro max) details updated.
```

Figure 7

Test Case 09

Enter a wrong item code for ‘UID’ (exception handling)

```
          • Type AID for adding item details.
          • Type DID for deleting item details.
          • Type UID for updating item details.
          • Type VID for viewing the items table.
          • Type SID for saving the item details to the text file at any time.
          • Type SDD for selecting four dealers randomly from a file.
          • Type VRL for displaying all the details of the randomly selected dealers.
          • Type LDI for display the items of the given dealer.
          • Type ESC to exit the program.

Enter your choice: UID
Enter Item Code: 0010
0010 not found in inventory.
```

Figure 8

Test Case 10

Enter ‘VID’ for view item

```
        • Type AID for adding item details.
        • Type DID for deleting item details.
        • Type UID for updating item details.
        • Type VID for viewing the items table.
        • Type SID for saving the item details to the text file at any time.
        • Type SDD for selecting four dealers randomly from a file.
        • Type VRL for displaying all the details of the randomly selected dealers.
        • Type LDI for display the items of the given dealer.
        • Type ESC to exit the program.

Enter your choice: VID
```

Item code	Item Name	Item Brand	Item Price	Item Quantity	Item Category	Purchased Date
0003	S22 Ultra	Samsung	310000	25	Mobile Phone	23.05.2023
0002	VIVA BOOK	ASUS	380000	30	LAPTOP	12.09.2023
0001	Iphone 14 Pro max	Apple	590000	20	Mobile Phone	12.11.2022

```
Total Purchased Items: 75
```

Figure 9

Test case 11

Enter ‘SID’ for saving the item details to the text file at any time.

```
        • Type AID for adding item details.
        • Type DID for deleting item details.
        • Type UID for updating item details.
        • Type VID for viewing the items table.
        • Type SID for saving the item details to the text file at any time.
        • Type SDD for selecting four dealers randomly from a file.
        • Type VRL for displaying all the details of the randomly selected dealers.
        • Type LDI for display the items of the given dealer.
        • Type ESC to exit the program.

Enter your choice: SID
Inventory saved.
```

Figure 10

Text File

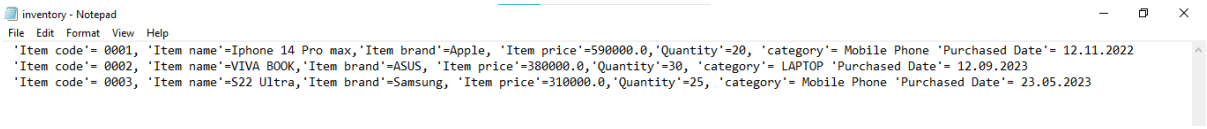


Figure 11

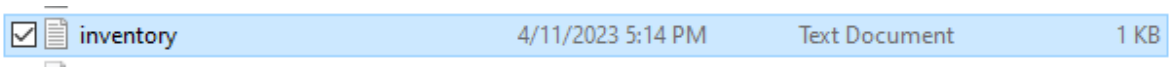


Figure 12

Text Case 12

Enter ‘SDD’ for selecting four dealers randomly from a file.

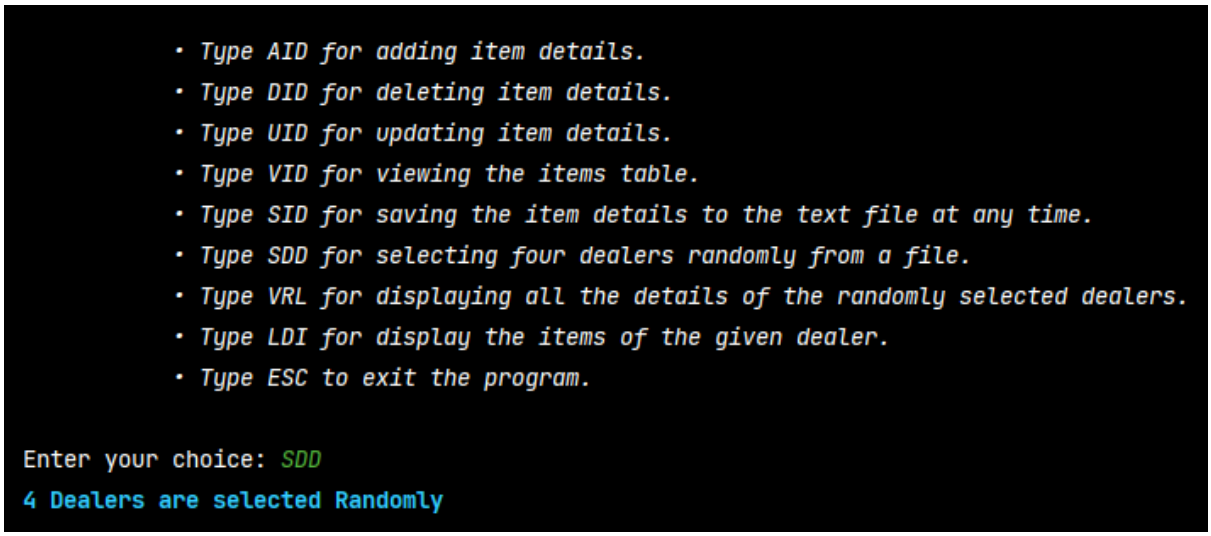


Figure 13

Test case 13

Enter “VRL” for displaying all the details of the randomly selected dealers.

```
Enter your choice: VRL
Dealer Name            Contact Number  Location    Item Name           Brand      price            Quantity
--------------------  ---------------  ----------  ------------------  ---------  -------------  ----------
Gawesh Gomes             +94761328236  Colombo
                                                   dell XPS Laptop     DELL       RS. 590000.00          50
                                                   Gaming keyboard     asus       RS. 4900.00            30
                                                   CPU                 DELL       RS. 220000.00          20
Radil Damsara            +94714325437  Kaluthara
                                                   flash drive         HP         Rs. 12000.00           40
                                                   Dell XPS 13         Dell       Rs. 550000.00          15
                                                   heat fan pro        Dell       Rs. 45000.00           5
Malindu_Dilshan          +94772534657  Panadura
                                                   Memory card         Sandisk    Rs. 4000.00            20
                                                   Memory card readers Ugreen     Rs. 1000.00            45
                                                   projector           ViewSonic  Rs. 120000.00          18
Sehandu_Siriwardhana     +94786756453  kurunagala
                                                   laptop bags         asus       Rs. 2500.00            50
                                                   Headphone           samsung    Rs. 3000.00            24
                                                   SSD                 Sandisk    Rs. 10000.00           35
```

Figure 14

Test Case 14

If User enters ‘VRL’ option without entering ‘SDD’ option (exception handling)

```
          • Type AID for adding item details.
          • Type DID for deleting item details.
          • Type UID for updating item details.
          • Type VID for viewing the items table.
          • Type SID for saving the item details to the text file at any time.
          • Type SDD for selecting four dealers randomly from a file.
          • Type VRL for displaying all the details of the randomly selected dealers.
          • Type LDI for display the items of the given dealer.
          • Type ESC to exit the program.

Enter your choice: VRL
You have not selected any dealers yet!!!
```

Figure 15

Test case 15

Enter ‘LDI’ for display the items of the given dealer.

```
          • Type AID for adding item details.
          • Type DID for deleting item details.
          • Type UID for updating item details.
          • Type VID for viewing the items table.
          • Type SID for saving the item details to the text file at any time.
          • Type SDD for selecting four dealers randomly from a file.
          • Type VRL for displaying all the details of the randomly selected dealers.
          • Type LDI for display the items of the given dealer.
          • Type ESC to exit the program.

Enter your choice: LDI
Enter Dealer Name ( Please select from the randomly selected dealer table ) : Diyathma_wijewardhana
Dealer name:  Diyathma_wijewardhana
```

Item Name	Brand	Quantity	Price
Iphone 14 pro max	Apple	10	Rs. 649000.00
computer lamp	orange	50	Rs. 4800.00
Iphone charger	apple	40	Rs. 12000.00

Figure 16

Test case 16

Enter a wrong name what not in the system (exception handling)

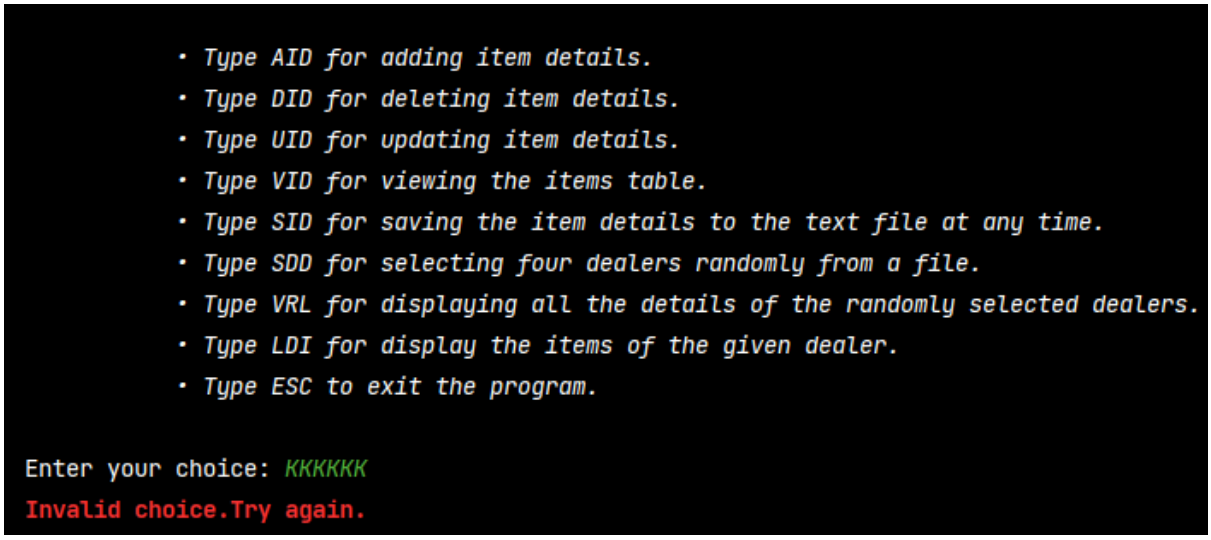


Figure 17

Test case 17

Enter ‘ESC’ to exit the program.

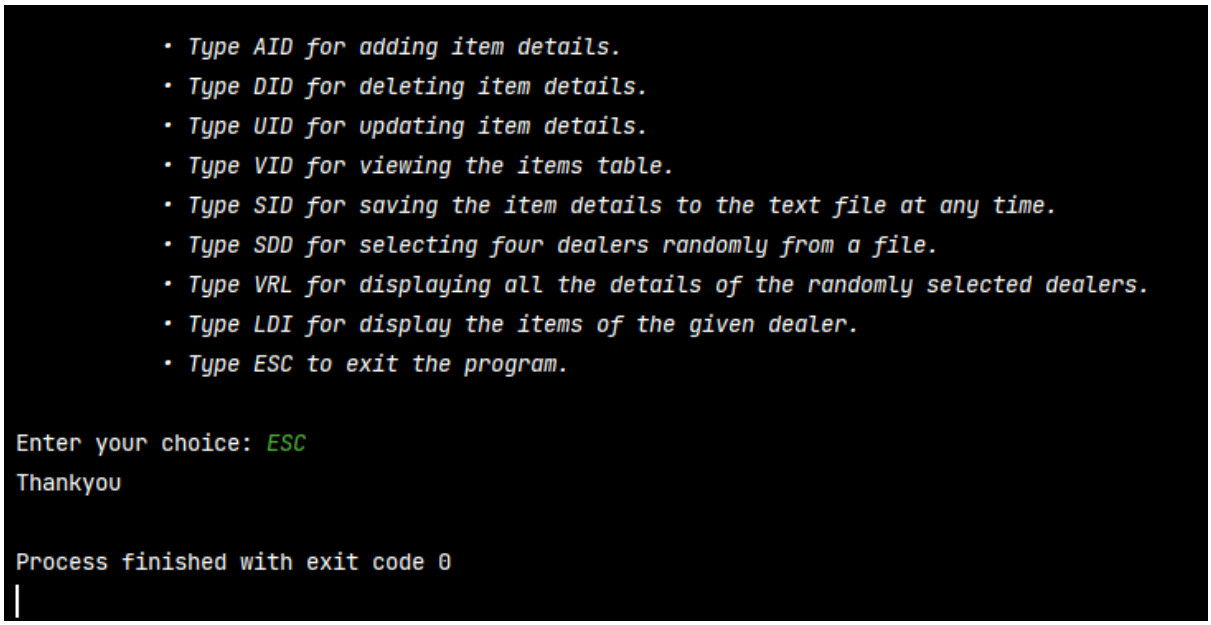


Figure 18

Conclusion

A One Net Cafe's inventory system is seen in the code above. The user can see, edit, add, and delete things from the inventory using the system. The system is built on a dictionary structure, in which every item is represented by a special code and a list of properties including name, brand, price, quantity, category, and date of purchase.

A straightforward command-line interface provided by the application helps the user navigate through the many procedures. It is interactive and user-friendly. For the purpose of preventing improper user input, the application also has systems for managing errors and validating data. The program then has a feature that displays the information for four dealers after randomly selecting them from a file. The program gains some entertainment value from this feature by adding an element.

Overall according to my assumption, the inventory system is a well-structured and functional program that provides an efficient and convenient way to manage the inventory of a One Net Cafe. The code can be further improved by adding more features, such as search or sorting functions, or by implementing a graphical user interface to enhance the user experience.