This file contains functions that will be used by other functions in the inventory system there by the name utilities

In [ ]:

```
#To be able to call this objects in other notebooks
import import_ipynb
```

In [ ]:

```
from categories import category_items,categories
import json
from datetime import date
```

In [ ]:

```
#A function to format the product names being provided to
#  manner that relates to the inventories
def productNameFormatter(productName: str):
    #if the product name is a two letter word format it is splitted then combined
    if(" " in productName):
        productName = productName.split(" ")
        productName = productName[0].lower() + productName[1].title()
        return productName
    else:
        productName = productName.lower()
        return productName
```

In [ ]:

```
# A function to check if a product exists in the inventories
def prodIsExist(prod):
    with open("inventories.json","r") as handler:
        data = json.load(handler)

    for key, value in category_items.items():

        if prod in value:
            if key == 'Beverages':
                return [key, data["beverages_inventory"][prod]]
            elif key == 'PhoneAccessories':
                return [key, data["phoneaccessories_inventory"][prod]]
            elif key == 'Toiletries':
                return [key, data["toiletries_inventory"][prod]]
            elif key == 'Pastry':
                return [key, data["pastry_inventory"][prod]]
            elif key == 'Cosmetics':
                return [key, data["cosmetics_inventory"][prod]]

    return False
```

In [ ]:

```
# A function that updates the sales records json file by adding
# the details of products purchased by customers there.
def addToSalesRecord(quantity, result, sales_price, productName,total_cost):
    sale_details = {}

    sale_details["quantity"] = quantity
    sale_details["productPrice"] = result[1]["purchasePrice"]
    sale_details["soldFor"] = sales_price
    sale_details["category"] = result[0]
    sale_details["productName"] = productName
    sale_details["date"] = f"{date.today()}"
    sale_details["totalCost"] = total_cost

    try:
        with open("sales_records.json", "r") as handler:
            salesRecords = json.load(handler)

        id = str(len(salesRecords) + 1)

        with open("sales_records.json", "w") as handler:
            salesRecords[id] = sale_details
            json.dump(salesRecords, handler, indent=2)

        return [True,"Sales record updated\n"]
    except:
        return [False,"FILE ERROR\n"]
```

In [ ]:

```
#A function to deduct the number of products bought from the inventories
# and generate an alert if the number of products falls below 5
def reduceInventoryQuantity(prod, qty):
    with open("inventories.json","r") as handler:
        data = json.load(handler)

    for key, value in category_items.items():
        if prod in value:
            if key == 'Beverages':
               data["beverages_inventory"][prod]["quantity"] = int(data["beverages_inventory"][prod]["quantity"]) - qty

               if data["beverages_inventory"][prod]["quantity"] < 5:
                print(f"[ALERT] {prod} NOW AS LESS THAN 5 AVAILABLE UNITS")

            elif key == 'PhoneAccessories':
                data["phoneaccessories_inventory"][prod]["quantity"] = int(data["phoneaccessories_inventory"][prod]["quantity"]) - qty

                if data["phoneaccessories_inventory"][prod]["quantity"] < 5:
                    print(f"[ALERT] {prod} NOW AS LESS THAN 5 AVAILABLE UNITS")

            elif key == 'Toiletries':
                data["toiletries_inventory"][prod]["quantity"] = int(data["toiletries_inventory"][prod]["quantity"]) - qty

                if data["toiletries_inventory"][prod]["quantity"] < 5:
                    print(f"[ALERT] {prod} NOW AS LESS THAN 5 AVAILABLE UNITS")

            elif key == 'Pastry':
                data["pastry_inventory"][prod]["quantity"] = int(data["pastry_inventory"][prod]["quantity"]) - qty

                if data["pastry_inventory"][prod]["quantity"] < 5:
                    print(f"[ALERT] {prod} NOW AS LESS THAN 5 AVAILABLE UNITS")

            elif key == 'Cosmetics':
                data["cosmetics_inventory"][prod]["quantity"] = int(data["cosmetics_inventory"][prod]["quantity"]) - qty

                if data["cosmetics_inventory"][prod]["quantity"] < 5:
                    print(f"[ALERT] {prod} NOW AS LESS THAN 5 AVAILABLE UNITS")

    with open("inventories.json", "w") as handler:
        json.dump(data,handler, indent=2)
```

In [ ]:

```
#A function used to generate receipts for purchases made by customers
def receiptGenerator(receipt):
    total = 0
    if len(receipt) > 0:
        print("RECEIPT")
        print("---------")
        for i in receipt:
            print(f"Product: {i['productName']} Quantity: {i['quantity']} subTotal: {i['total']}")
            total = total + i['total']
        print(f"Total: {total}")
    else:
        print("YOUR RECEIPT IS EMPTY")
```

In [ ]:

```
#A function that answers the option b,c,d of requirement 6 as stated in the pdf
def onEachSales():
    with open("sales_records.json", "r") as handler:
        salesRecords = json.load(handler)

    #To show total sales for each categories
    print("TOTAL SALES BY CATEGORY")
    print("_______________________")
    for category in categories:
        catTotal = 0
        for record in salesRecords.values():
            if category == record["category"]:
                catTotal += record["totalCost"]
        print(f"{category} ==> {catTotal}")

    #To show total sales for each product
    print("\n")
    print("TOTAL SALES BY PRODUCT")
    print("_______________________")

    newarray = []
    for i in category_items.values():
        newarray += i
    for products in newarray:
        productTotal = 0
        for record in salesRecords.values():
            if products == record["productName"]:
                productTotal += record["totalCost"]
        print(f"{products} ==> {productTotal}")

    #To show total sales for each date recorded in the sales record json file
    print("\n")
    print("TOTAL SALES BY DATE")
    print("_______________________")
    #I used sets to remove repetetions
    dateArray = set()
    for i in salesRecords.values():
        dateArray.add(i["date"])
    for date in dateArray:
        productTotal = 0
        for record in salesRecords.values():
            if date == record["date"]:
                productTotal += record["totalCost"]
        print(f"{date} ==> {productTotal}")
```