Inventory Management system For Large Scale Industries

Team Members:

1. Sarvesh Pathak
   sarvesh.20bcr7058@vitap.ac.in
2. Ch.John Abhishek
   abhishek.18bev7011@vitap.ac.in
3. M.Sai Varshith
   Varshith.18bec7119@vitap.ac.in
4. B Sravan Kumar
   sravan.18bec7053@vitap.ac.in

Introduction

a.Overview:

In this project ,the python code is written in such a way that it scans the qr code and gets the product details from it .we are using cloudant DB as a database to store the product details.The product details can be seen in cloudant.we have configured the Node-RED flow to receive data from IBM IoT platform.A web app is created to visualize the stock inward and outward details.

b. Purpose:

The main purpose of inventory management is to help businesses easily and efficiently manage the ordering, stocking, storing and using of inventory. By effectively managing your inventory, you'll always know what items are in stock, how much of them there are, and where they are located. An inventory management system (or inventory system) is the process by which you track your goods throughout your entire supply chain, from purchasing to production to end sales. It governs how you approach inventory management for your business.Inventory management systems are widely used in a variety of industries, from manufacturing to utilities, healthcare, education, government, and more.It is the combination of technology (hardware and software) and processes and procedures that oversee the monitoring and maintenance of stocked products, whether those products are company assets, raw materials and supplies, or finished products ready to be sent to vendors or end consumers.In some industries, inventory management is also known as stock management.

2. Literature Survey

a. Existing problem

Some of the challenges of inventory management are mentioned below:

1.Inconsistent Tracking:

Using manual inventory tracking procedures across different software and spreadsheets is time-consuming, redundant and vulnerable to errors. Even small businesses can benefit from a centralized inventory tracking system that includes accounting features.

2.Changing Demand:

Customer demand is constantly shifting. Keeping too much could result in obsolete inventory you're unable to sell, while keeping too little could leave you unable to fulfill customer orders. Order strategies for core items, as well as technology to create and execute an inventory plan, can help compensate for changing demand.

3.Manual Documentation:

Managing inventory with paperwork and manual processes is tedious and not secure. And it doesn't easily scale across multiple warehouses with lots of stock.

4.Poor Communication:

Communication and collaboration are key. When departments are apathetic about sharing information, it makes identifying inventory trends and finding ways to improve much more difficult.

5.Overstocking:

Keeping too much stock on hand can be as problematic as having too little. Overstock impacts business cash flow and leads to inventory-related problems, such as storage and loss.

6.Inadequate Software:

To scale inventory management software to support complex logistics, it needs to integrate with your existing business process platforms. The difficult task is choosing from hundreds of inventory management solutions and mastering a host of features that require training and ongoing support.

b. Proposed solution

1.QRcode or quick response codes are advanced than barcodes .QRcodes occupy less space than barcodes.Unlike barcodes if a QR code is damaged up to 30% it can still be scannable.Barcodes are unidirectional ,QR codes can be scanned from any direction.
2.use of web app makes easier to look at the details of the stock coming in and going out.

3. Theoretical Analysis

a. Block diagram:

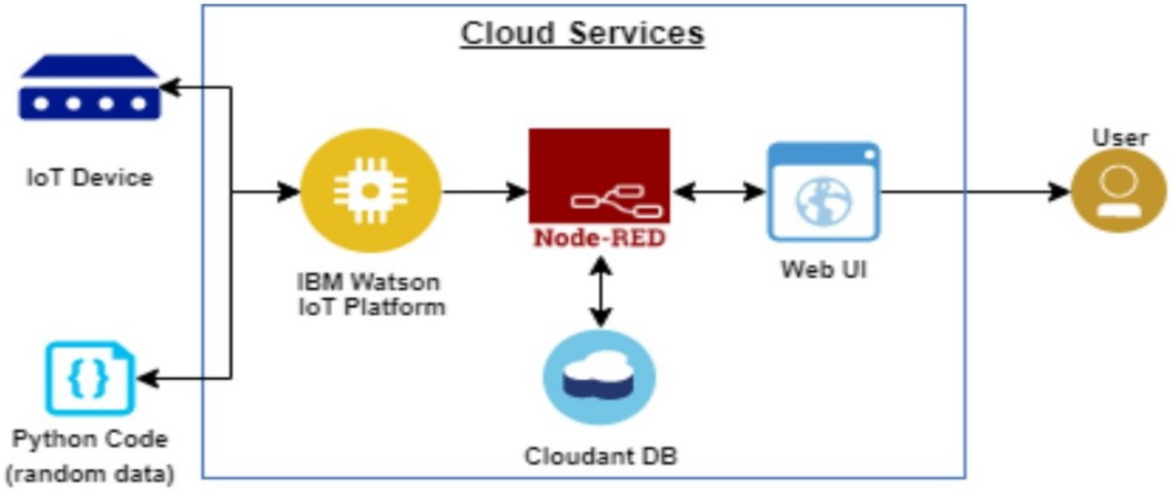
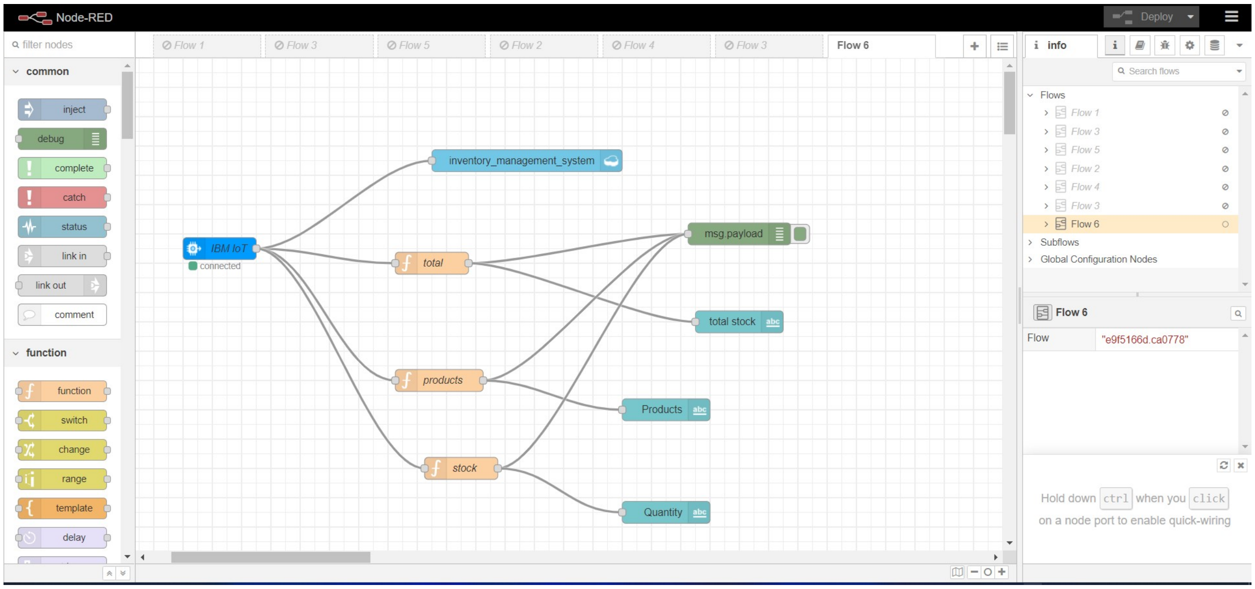


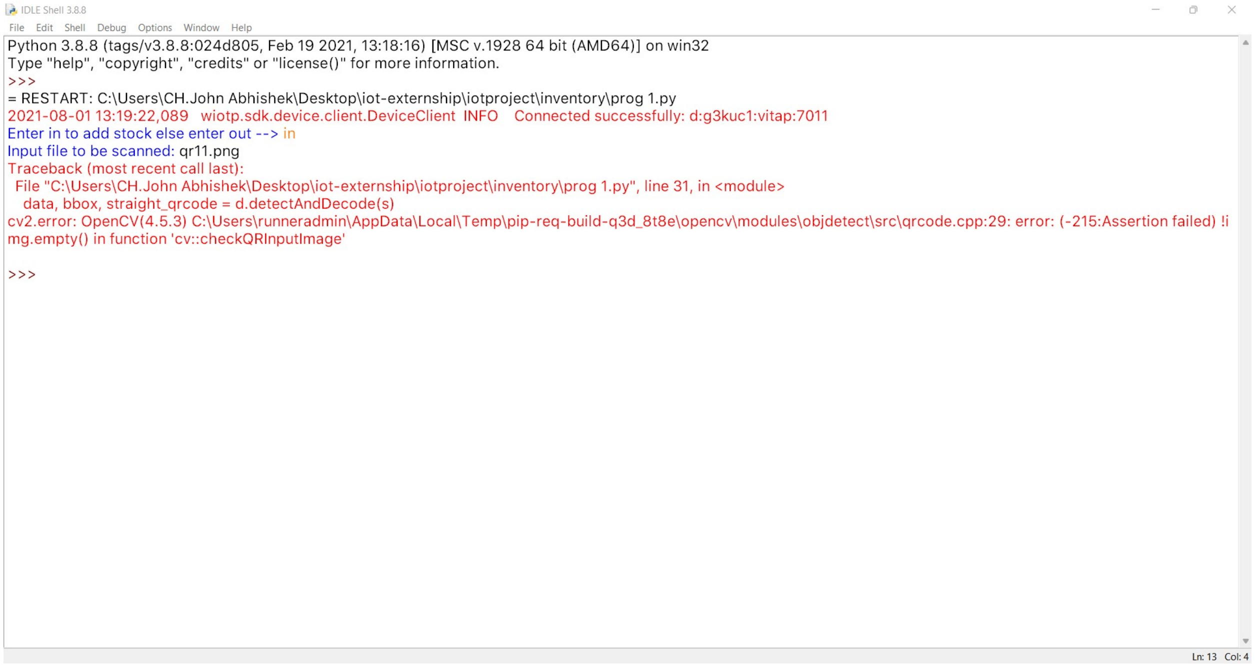
Fig 1:Block diagram

b. Hardware / Software designing:



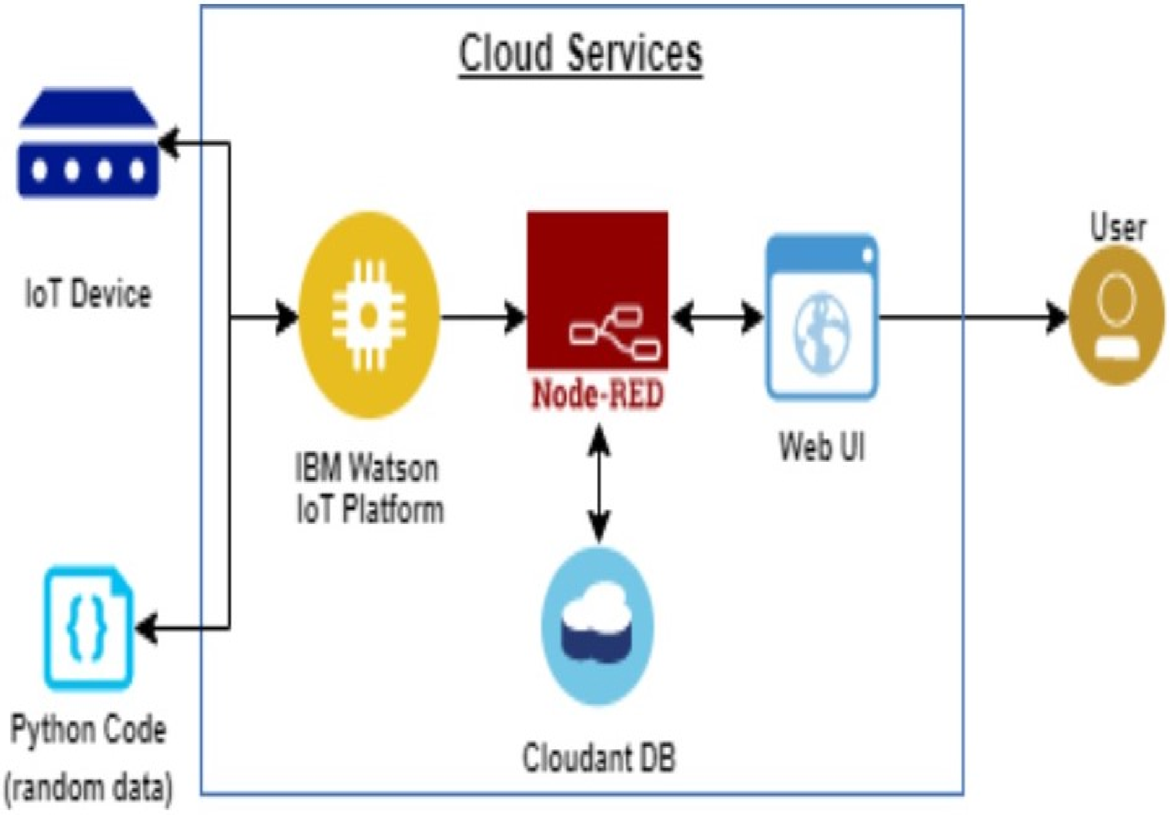
4. Experimental Investigations:

when the qr code is not able to be scanned or if there is an error in the qr code then it is giving an error

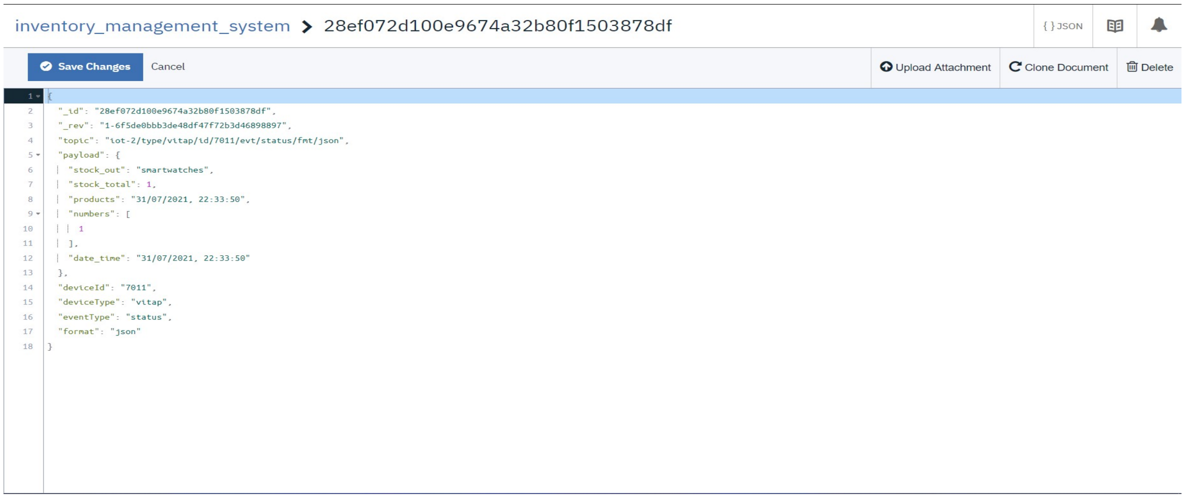


```
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\CH.John Abhishek\Desktop\iot-externship\iotproject\inventory\prog 1.py
2021-08-01 13:19:22,089   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:g3kuc1:vitap:7011
Enter in to add stock else enter out --> in
Input file to be scanned: qr11.png
Traceback (most recent call last):
  File "C:\Users\CH.John Abhishek\Desktop\iot-externship\iotproject\inventory\prog 1.py", line 31, in <module>
    data, bbox, straight_qrcode = d.detectAndDecode(s)
cv2.error: OpenCV(4.5.3) C:\Users\runneradmin\AppData\Local\Temp\pip-req-build-q3d_8t8e\opencv\modules\objdetect\src\qrcode.cpp:29: error: (-215:Assertion failed) !i
mg.empty() in function 'cv::checkQRInputImage'

>>>
```

5. Flowchart



6. Result:



```
{
  "_id": "28ef072d100e9674a32b80f1503878df",
  "_rev": "1-6f5de0bbb3de48df47f72b3d46898897",
  "topic": "iot-2/type/vitap/id/7011/evt/status/fmt/json",
  "payload": {
    "stock_out": "smartwatches",
    "stock_total": 1,
    "products": "31/07/2021, 22:33:50",
    "numbers": [
      1
    ],
    "date_time": "31/07/2021, 22:33:50"
  },
  "deviceId": "7011",
  "deviceType": "vitap",
  "eventType": "status",
  "format": "json"
}
```

Details of the products are stored in the cloudant database

7. Advantages & Disadvantages:

Advantages:

1. **It leads to a more organized warehouse:** With the help of a good inventory management system, you can easily organize your warehouse. If your warehouse is not organized, you will find it very difficult to manage your inventory. A lot of businesses choose to optimize their warehouse by putting the items that have the highest sales together in a place that is easy to access in the warehouse.
3. **It saves time and money:** an effective inventory management system can translate to time and money saved on the part of the business. By keeping track of the product that you already have at hand, you can save yourself the hassles of having to do an inventory recount in order to ensure your records are accurate. It also allows you to save cash that would have otherwise been spent on slow moving products.
4. **Improves efficiency and productivity:** inventory management devices like bar code scanners and inventory management software can help to greatly increase the efficiency and productivity of a business. They do this by eliminating the manual way of doing things thus allowing employees to do other more important things for the business.

Disadvantages of inventory management:

Disadvantages:

1. **Production problem:** even though inventory management can reveal to you the amount of stock you have at hand and the amount that you have sold off, it can also hide production problems that could lead to customer service disasters. Since the management places almost all of its focus on inventory management to the detriment of quality control, broken or incorrect items that would normally be discarded

are shipped along with wholesome items.

2. **High implementation costs:**some inventory management systems can come at a high price because the business needs to install specialized systems and software in order to use them. This can be problematic for large businesses which operate in difficult locations. Even after installing the costly system, it still needs to be maintained and upgraded on a regular basis, thus incurring more costs.
3. Even with an efficient inventory management method, you can control but not eliminate business risk.
4. The control of inventory is complex because of the many functions it performs. It should thus be viewed as a shared responsibility.

8. Applications:

1. Companies use inventory management software to avoid product overstock and outages.
2. It also helps companies keep lost sales to a minimum by having enough stock on hand to meet demand.
3. Tracking where products are stocked, which suppliers they come from, and the length of time they are stored is made possible with inventory management software. By analysing such data, companies can control inventory levels and maximize the use of warehouse space.

9.Conclusion:

Inventory management system can help you save money and keep an accurate stock count.When you automate some warehouse operations, it’s easy to skip a physical inventory check.The cost of inventory management software can seem daunting to a small business,though its helps in increasing the profit.

10. Future Scope:

Multilocation management: Manage multiple warehouses and points-of-sale (POS). All locations can be integrated within a single inventory management system.

Stock notifications: Receive alerts and notifications when there's over- or under-stocking . This helps you to place orders or offer promotional discounts to clear out extra stock.

Report generation: View sales history in the form of a list of your most popular products. This feature also enables you to manage items in your inventory that have not reached the sales levels you expected.

Warehouse management: This feature is useful if you need to optimize your warehouse stock and maintain an accurate log of each product's location. It'll give you a single view of where all of your products are stored.

Stock returns handling: Manage returns more effectively by reducing time-to-return through automation of the entire process.

11. Bibliography:

- https://blog.capterra.com/what-is-an-inventory-management-system/
- https://en.wikipedia.org/wiki/Inventory_management_software#Advantages_of_ERP_inventory_management_software

12. Appendix

a. Source code

python code for generating qrcode:

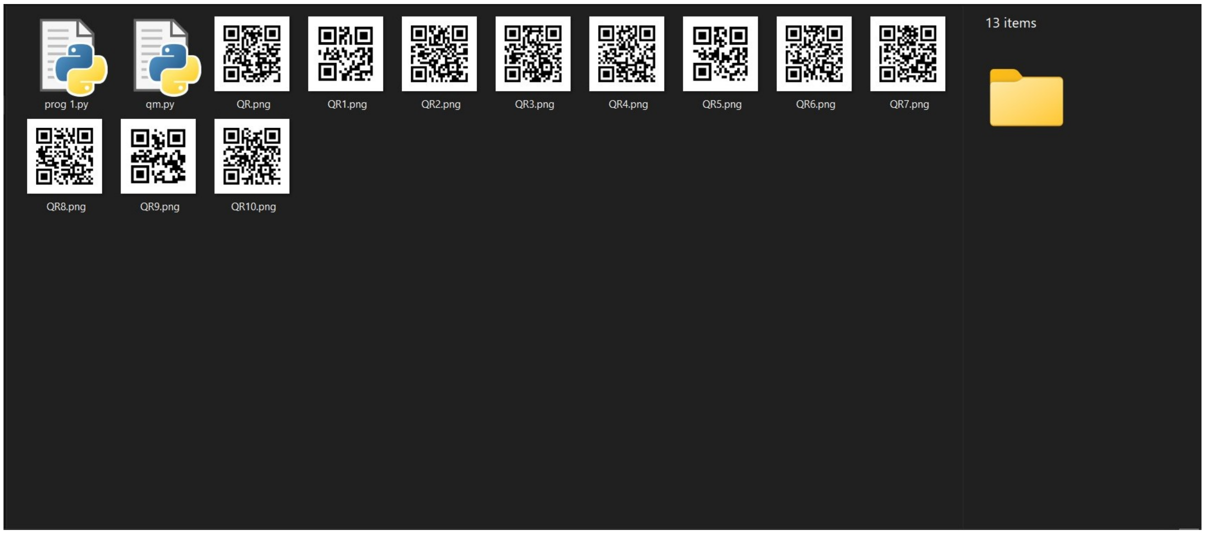
```
import pyqrcode
import cv2
data="mobilephones"
data1="tablets"
data2="headphones"
data3="smartwatches"
data4="speakers"
data5="laptops"
data6="accessories"
data7="earphones"
data8="Smart TV"
data9="cameras"
data10="Home Theatre"


my_QR =pyqrcode.create(data)
my_QR1 =pyqrcode.create(data1)
my_QR2 =pyqrcode.create(data2)
my_QR3 =pyqrcode.create(data3)
my_QR4=pyqrcode.create(data4)
my_QR5 =pyqrcode.create(data5)
my_QR6 =pyqrcode.create(data6)
my_QR7 =pyqrcode.create(data7)
my_QR8 =pyqrcode.create(data8)
my_QR9 =pyqrcode.create(data9)
my_QR10 =pyqrcode.create(data10)

my_QR.png("QR.png", scale="10")
my_QR1.png("QR1.png", scale="10")
my_QR2.png("QR2.png", scale="10")
my_QR3.png("QR3.png", scale="10")
my_QR4.png("QR4.png", scale="10")
my_QR5.png("QR5.png", scale="10")
my_QR6.png("QR6.png", scale="10")
my_QR7.png("QR7.png", scale="10")
my_QR8.png("QR8.png", scale="10")
my_QR9.png("QR9.png", scale="10")
my_QR10.png("QR10.png", scale="10")


s=cv2.imread('QR.png')
s1=cv2.imread('QR1.png')
```

```
s2=cv2.imread('QR2.png')
s3=cv2.imread('QR3.png')
s4=cv2.imread('QR4.png')
s5=cv2.imread('QR5.png')
s6=cv2.imread('QR6.png')
s7=cv2.imread('QR6.png')
s8=cv2.imread('QR6.png')
s9=cv2.imread('QR6.png')
s10=cv2.imread('QR6.png')
d=cv2.QRCodeDetector()
data, bbox, straight_qrcode = d.detectAndDecode(s)
data, bbox, straight_qrcode = d.detectAndDecode(s1)
data, bbox, straight_qrcode = d.detectAndDecode(s2)
data, bbox, straight_qrcode = d.detectAndDecode(s3)
data, bbox, straight_qrcode = d.detectAndDecode(s4)
data, bbox, straight_qrcode = d.detectAndDecode(s5)
data, bbox, straight_qrcode = d.detectAndDecode(s6)
data, bbox, straight_qrcode = d.detectAndDecode(s7)
data, bbox, straight_qrcode = d.detectAndDecode(s8)
data, bbox, straight_qrcode = d.detectAndDecode(s9)
data, bbox, straight_qrcode = d.detectAndDecode(s10)

print(d)
if bbox is not None:
 print(data)
```

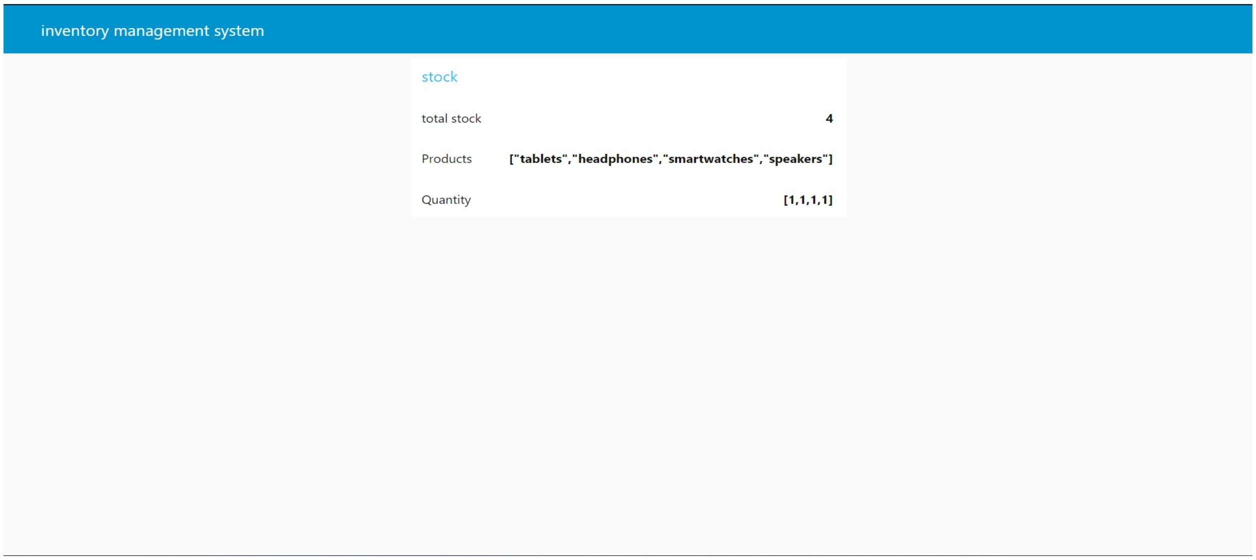
Python code for inventory system:

```
import wiotp.sdk.device
import datetime
import time

import pyqrcode
import cv2
myConfig = {
    "identity": {
        "orgId": "g3kuc1",
        "typeId": "vitap",
        "deviceId":"7011"
    },
    "auth": {
        "token": "vitap123"
    }
}

def myCommandCallback (cmd):
    print ("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient (config=myConfig, logHandlers=None)
client.connect()
i=0
p=[]
n=[]
c=input("Enter in to add stock else enter out --> ")
while True:
 s=cv2.imread(input("Input file to be scanned: "))
 d=cv2.QRCodeDetector()
 data, bbox, straight_qrcode = d.detectAndDecode(s)
 r=datetime.datetime.now().strftime("%d/%m/%Y, %H:%M:%S")
 if c=='in':
  if(p.count(data)>=1):
     n[p.index(data)]=n[p.index(data)]+1
     i=i+1
  if(p.count(data)==0):
     p.append(data)
     n.append(1)
     i=i+1
  if bbox is not None:
    myData={'stock_in': data,'stock_total': i,'products':p ,'numbers':n, 'date_time':r }
    client.publishEvent (eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print ("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
 else :
  if(p.count(data)>=1):
     i=i-1
     n[p.index(data)]=n[p.index(data)]-1
  if bbox is not None:
   myData={'stock_out': data, 'stock_total': i, 'products':r ,'numbers':n ,'date_time':r}
   client.publishEvent (eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
   print ("Published data Successfully: %s", myData)
  client.commandCallback = myCommandCallback
 c=input("Enter in to add stock else enter out --> ")

client.disconnect()
```

b. UI output Screenshot.



QR codes are generated in the project folder



inventory management system

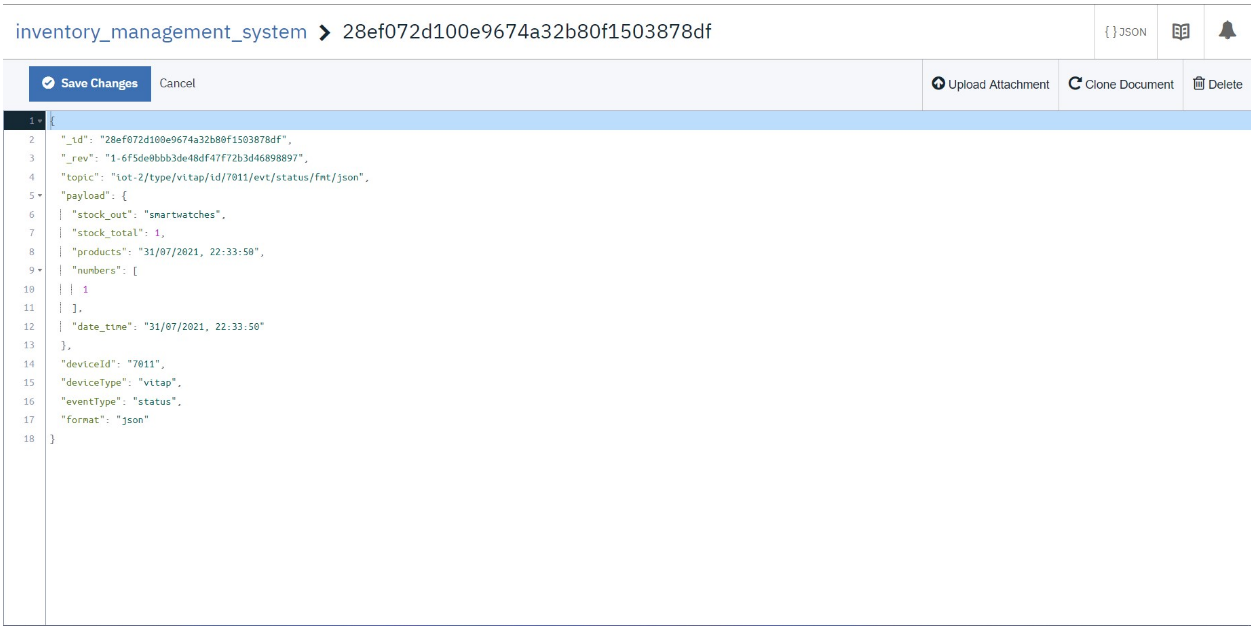
stock

total stock 4

Products ["tablets","headphones","smartwatches","speakers"]

Quantity [1,1,1,1]

Web ui.



```
{
  "_id": "28ef072d100e9674a32b80f1503878df",
  "_rev": "1-6f5de0bbb3de48df47f72b3d46898897",
  "topic": "iot-2/type/vitap/id/7011/evt/status/fmt/json",
  "payload": {
    "stock_out": "smartwatches",
    "stock_total": 1,
    "products": "31/07/2021, 22:33:50",
    "numbers": [
      1
    ],
    "date_time": "31/07/2021, 22:33:50"
  },
  "deviceId": "7011",
  "deviceType": "vitap",
  "eventType": "status",
  "format": "json"
}
```

Every time after reading QR code the product details along with the codes are sent to the cloud platform with date and time.