Inventory Management System for Large Scale Industries

By Batch no-A08:

1.Chirakala. Sai Malavika

2.Choragudi. Mamatha

3.Chori. Nirmala

4.Choutapalli. Viswa Bharathi

5.Dara. Krishna Vamsi

Table of Contents:

1.Introduction

a. Overview:

The project Inventory Management System is a complete laptop web-cam based application designed on python programming language. The main aim of the project is to develop Inventory Management System Model software in which all the information regarding the stock of the organization will be presented.Inventory management system which is a QR based system, plays a significant role in the management of stocks & Maintenance of equipment's for any or - Inventory Management System is software which is helpful for the businesses operate hardware stores, where store-owner keeps the records of stock inventory. This project eliminates the paper work, human faults, manual delay and speed up process. Inventory Management System is an application developed for operating systems which focused in the area of Inventory control and generates the various required report organization.

b. Purpose:

The purpose of this "QR Based Stock Management System" software project is same as it name, i.e., it is used for the recording or scanning the Stock related information. It is developed to manage the stock information, so that people who work in the organization can access accurate stock information quickly and easily as and when required, thereby improving its operational efficiency and effectiveness. Computerized software system helps to fulfil these goals. Computerization of the official work will help in doing a lot of manual work quickly. It will help in easy storage and access of all information, in short period of time. The development of this software access facilitates accurate information correctly and easily which leads to increase efficiency & effectiveness of the organization too. This project reduces the amount of work the workers have to do. The works will not have to write item related information on a tag manually. The workers will not have to go to owner and ask information related to the item, and they also don't have to calculate the item price, as it will be automatically calculated. The entire process of stock keeping using our system reduces the probability of human error and reduces the time required for the entire process of stock maintenance.

2.Literature Survey

a. Existing problems:

The existing Stock Management procedure is a manual one which run with pen and paper. It is very time consuming and uneconomical for an organization. Data organization is not effective and efficient in the current system, and there is more risk of data mismanagement. There are number of Inventory Management System available in the market. After doing my research, I have come to know that most of them are limited to few products. Some others are lacking in good UI. Marketing points are not much focus to increasing sales. Customer management system and Inventory Management system can't be linked due to different organization which leads to compromising the client satisfaction level. Most of them are not using cloud computer concept but we are trying to develop such a system which is for everyone rather than for only big companies or for small organisation.

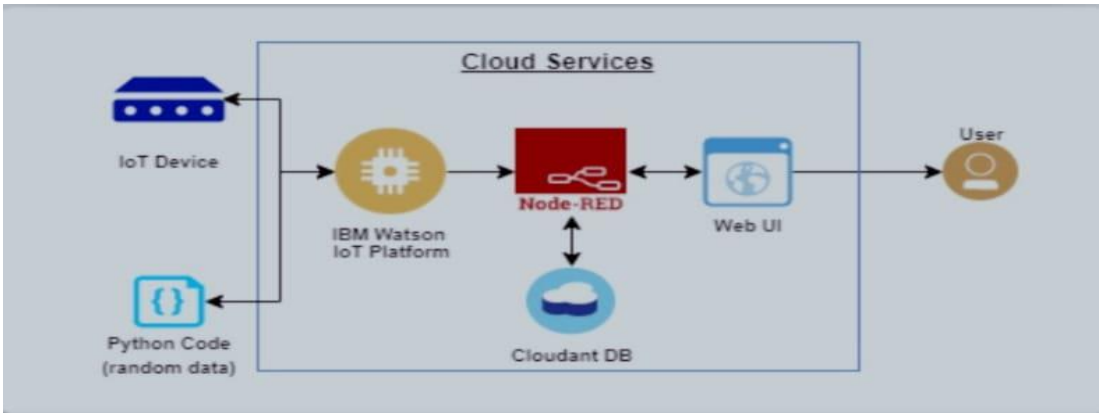
• Currently using pen & paper and excel sheets for maintaining the database.
• Time consuming, prone to error & cost intensive
•Requires more man power.

b. Proposed Solution:

The proposed system, which consists of a laptop web-cam application under the domain of Inventory Management System for any Inventory products. The Application utilizes QR (Quick Response) Code Technology. Using such a technology in which inventories can be maintained, manipulated easily and can be updated as well. The authorized user will scan the respective QR code on the device of product. The device will scan it and show all the quantity details of the product. The page will consist of data of stock details regarding the product.

3.Theoretical Analysis

a. Block diagram:



Explanation:

After running the python code, the web cam of our laptop scans the QR code of certain products and then modify the stock by adding or removing as we required that finally displays the list of stock maintained in our company or organisation.

b. Hardware/software designing:

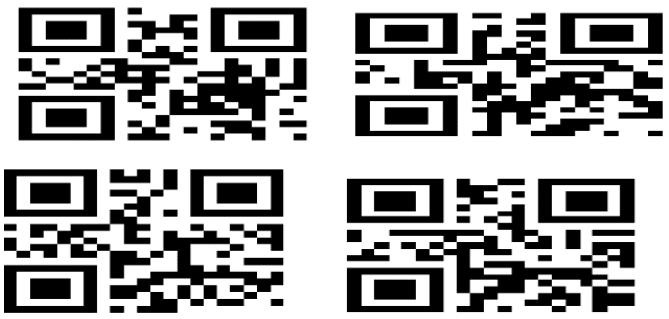
We follow a step-by-step procedure to set up all the interfaces required for our project and develop the code in python to list the stock of products in an organisation to the cloud. The following software is required:

- Python Idle (with specified packages installed)
- IBM cloud
- Node Red service

4.Experimental Investigations:

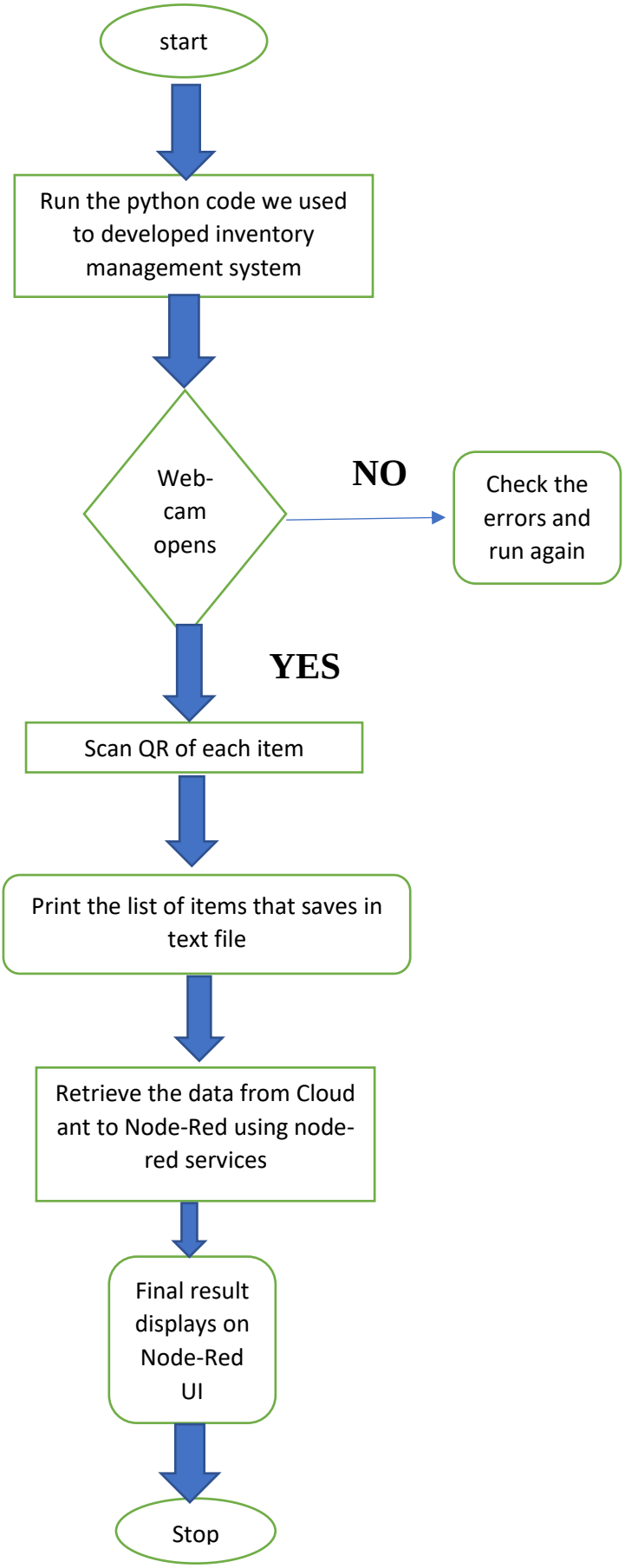
To successfully design this project QR code is generated. Generation of QR Code involves QR code can then be printed by Label printer which can be used to attach it to each item. Then, Open the QR Code Scanner, scans the QR Code which is attached to each item and after the QR Code is scanned the item related information will be shown on the screen which contains the: no of the product, and quantity of products. After scanning the QR Code, the authorized user will get complete detail of the product. In future, if any maintenance occurs that record can be updated using the Update option. The option is provided i.e., Scanning of QR Code using web-cam of our laptop. The Scan QR Code activity scans the QR Code and the information is then fetched from the database and is displayed on the same activity.

The displayed data is stored in IBM cloudant and using node-red services, a node-red flow is created which retrieves data from IBM cloud.

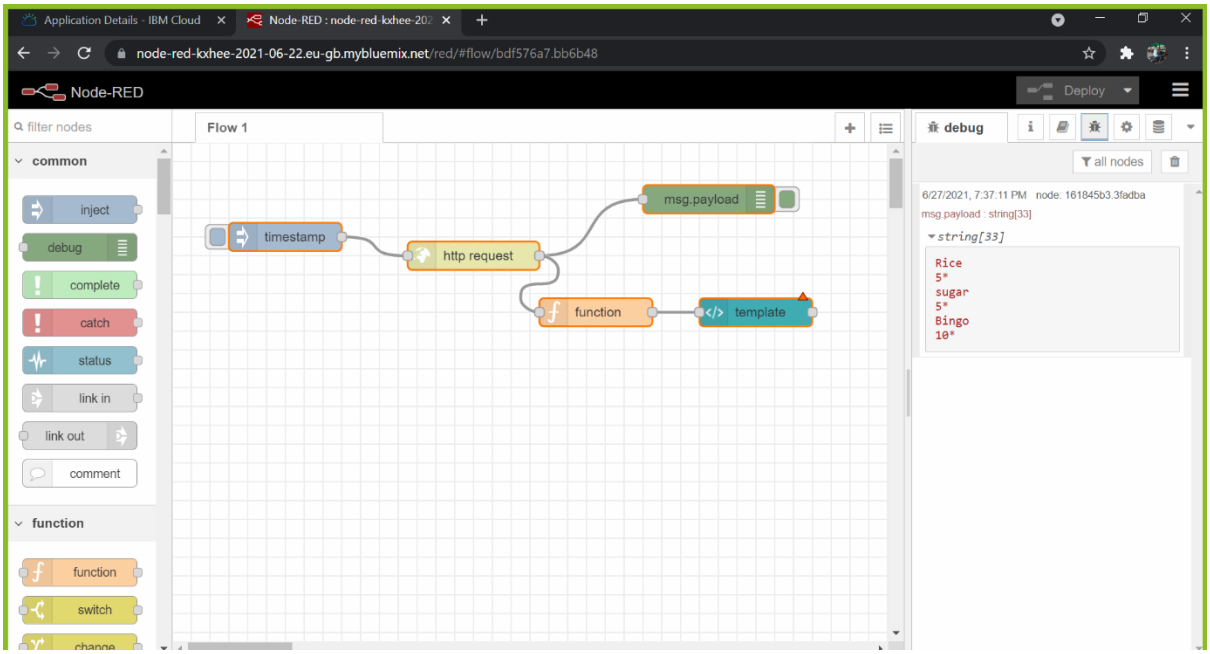


The Qr codes we used in this project

5.Flow chart:


start
Run the python code we used to developed inventory management system
Web-cam opens
NO
Check the errors and run again
YES
Scan QR of each item
Print the list of items that saves in text file
Retrieve the data from Cloud ant to Node-Red using node-red services
Final result displays on Node-Red UI
Stop

6.Result:



Explanation:

The result of our project fulfils our requirements. We have successfully Executed the designed program and the results have been as expected. This will help to reduce paperwork, time and human error. The method developed proves that it is effective and efficient method to manage and manipulate the Inventory Management System in Large Scale Industries.

7. Advantages:

- Scan products swiftly
- Easy to use
- Enhanced security as opposed to other technologies
- Minimal errors
- Quick automation
- Easy troubleshooting
- Elevated fault tolerance
- Superior data storage capabilities
- Dynamic to use

Disadvantages:

To shape this project it takes a special scanning device to read them, which can be a hefty expense for companies with large inventories.

8.Applications:

- Used in organisations for business purposes.
- Used in e-office inventory

9.Conclusion:

Inventory Management System Project is developed as a web-cam scanning application to meet the current stock management demands of a company or organization. The system can be accessed from anywhere with the internet.

10.Future Scope:

It is used to Inventory management system which is a QR based system, plays a significant role in the management of stocks & Maintenance of equipment's for any or - Inventory Management System is software which is helpful for the businesses operate hardware stores, where store-owner keeps the records of sales and purchase.

11.Bibliography:

https://code-projects.org/simple-inventory-management-in-python-with-source-code/
https://www.irjet.net/archives/V8/i4/IRJET-V8I4289.pdf

12.Appendix

a. Source code:

➢ Code for Managing Inventory:

```
import os
import fileinput
import sys
import cv2
import numpy as np
def abort():
    CHOICE = int(input('Enter 98 to continue or 99 to exit: ')) == 98
    menuDisplay() if CHOICE else sys.exit()
def menuDisplay():
    print("\n-->Welcome to Inventory management menu<--\n")
    print("(1) add new item")
    print("(2) remove item")
    print("(3) print list")
    print("(4) exit")
    CHOICE = int(input("Enter user choice : "))
    menuSelection(CHOICE)
def menuSelection(CHOICE):
    if CHOICE == 1:
        addInventory()
    elif CHOICE == 2:
        removeInventory()
    elif CHOICE == 3:
        printInventory()
    elif CHOICE != 4:
        print("Enter Valid Choice")
        menuDisplay()
    sys.exit()
def addInventory():
    print("scan your qr code when the camera asks")
    print("Adding Inventory")
    print("================")
    def video_reader():
```

```
    cam = cv2.VideoCapture(0)
      detector = cv2.QRCodeDetector()
      while True:
        _, img = cam.read()
        data, bbox, _ = detector.detectAndDecode(img)
        if data:
          print(data)
          with open('Inventory.txt', 'a') as f:
            f.write(data + '*\n')
          break
        cv2.imshow("img", img)
        if cv2.waitKey(1) == ord("Q"):
          break
      cam.release()
      cv2.destroyAllWindows()
    video_reader()
    abort()
def removeInventory():
    print("Removing Inventory")
    print("==================")
    def video_reader():
      cam = cv2.VideoCapture(0)
      detector = cv2.QRCodeDetector()
      while True:
        _, img = cam.read()
        data, bbox, _ = detector.detectAndDecode(img)
        if data:
          print(data)
          with open('Inventory.txt', 'r') as f:
            file = f.read()
          file = list(file.split("*\n"))
          file.remove(data)
          with open('Inventory.txt', 'w') as f:
            f.truncate()
            f.write("*\n".join(file))
          break
        cv2.imshow("img", img)
        if cv2.waitKey(1) == ord("Q"):
          break
      cam.release()
      cv2.destroyAllWindows()
    video_reader()
    abort()
def printInventory():
    with open('Inventory.txt', 'r') as f:
      InventoryFile = f.read()
    InventoryFile = list(InventoryFile.split("*\n"))
    print('Current Inventory')
    print('-----------------')
    for item in InventoryFile:
      if item != '':
        item_description, item_quantity = item.rstrip("\n").split("\n")
        print('Item:     ', item_description)
        print('Quantity: ', item_quantity.rstrip("*"))
        print('----------')
    abort()
```

```
menuDisplay()
```

- Code for uploading data in IBM Cloud:

```
import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError
import sys
import time
from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey


COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
#s3.us-south.cloud-object-storage.appdomain.cloud" # Current list avaiable at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints
COS_API_KEY_ID = "8MKJ8cj2FDFFxL5L9f4WE1HTq07nv-YRjJUT_g7eFp9A" # eg "W00YiRnLW4a3fTjMB-oiB-2ySfTrFBIQQWanc--P3byk"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/aeda64267f3b41b2878d07b5ba0aff74:1ea4ecb2-490a-44f7-b328-82be1e8428c8::"
# Create resource
cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_RESOURCE_CRN,
    ibm_auth_endpoint=COS_AUTH_ENDPOINT,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)

def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
```

```
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))


multi_part_upload("malavika1234","Inventory.txt","C:/Users/vosha/Desktop/inventory.txt")
```

b. UI output: