**Group 3**

**IOT Solution**

**IFS353**

**Due: 14 November 2023**

# Table of contents:

[**Table of contents: 2**](#_u5olp3pn8coz)

[**Introduction 3**](#_3u94timqb30q)

[**Challenges 3**](#_kcpq5r1e57a6)

[**Opportunities 4**](#_gqtwa664b08p)

[**Solution architecture 5**](#_5xuznf67tvnx)

[**Data flow 6**](#_fq296a8smoqy)

[**Code structure 7**](#_53lq0f9kbdvb)

[Temperature and humidity 7](#_yx22x8ee1ehb)

[Gas sensor 8](#_mhr06u12rjkn)

[GPS sensor 9](#_klv3j8ix2kko)

[RFID 9](#_wxb4vi4sc3n8)

[Camera and light sensor 10](#_de8rt9wl0u46)

[**Gantt chart 12**](#_hfzl90q00pt7)

[**What can be added/improved on? 12**](#_qr26k2h2n8m0)

[**Conclusion 12**](#_t5wabb3mgf1h)

[**Delegation of tasks 13**](#_dqpu1b4uc6ye)

[**Reference List 14**](#_badqfhlyb6hx)

## 

## 

## 

# Introduction

Good inventory management is essential to the success of an entity, better inventory management improves cash flows and reduces expenses. South Africa in recent months in 2023 have seen an increase in food prices, grabbing the attention of IoT specialists to come up with an IoT innovation to help reduce cost in the operations of goods manufactured. This assignment requires the use of an IOT system, and various sensors to achieve the IoT innovation. In this documentation an assessment of effectiveness of the IoT innovation in addressing the important challenges and opportunities in inventory management specifically with perishable goods (Asset tracking). It will be detailed by covering the challenges solved, opportunity arising, architectural solution, data flow, and the structure of the code of the IoT innovation.

# Challenges

The Sensor Selection process requires selecting the appropriate sensors for the monitoring of fruit behaviour and it is a fundamental challenge. To accurately capture changes during transit, multiple sensor types are required. Temperature and humidity sensors are essential to maintain fruit quality, while vibration sensors might detect any rough handling. Furthermore, gas concentration sensors are crucial for monitoring ripeness and spoilage. Each sensor choice carries its unique set of challenges, including calibration and integration into the Raspberry Pi system.

Power Management is another area of concern for the implementation process because the power consumption of Raspberry Pi devices is constant. In farming applications, access to a stable power source may be limited, which necessitates efficient power management. A goal here would be to balance operational requirements and power conservation to ensure that the Raspberry Pi can function for an extended period without frequent recharging or battery replacement.

Real-time data transmission from the Raspberry Pi can be challenging, especially in rural or remote farming areas. Connectivity options such as cellular networks, satellite communication, or Low-Power Wide-Area Networks (LoRaWAN) must be considered. Each option carries its associated costs and complexities, making it necessary to choose a solution that aligns with the specific farming environment.

Data Storage and Analysis, managing the vast amounts of data generated by the sensors is crucial. A system for storing, processing, and analysing this data must be designed. This may involve using cloud-based services or on-device storage and analytics, each with its pros and cons.

Data Accuracy, ensuring the accuracy of sensor readings is vital for making informed decisions. Sensors can drift over time and may require regular calibration. Establishing a comprehensive calibration and maintenance plan is essential to maintain data accuracy.

Data Integration challenges include integrating the data into existing farming systems and workflows which can become complex. Different farmers may be using a variety of tools and platforms, and it is imperative that the data provided by the Raspberry Pi system is compatible and useful to each of them.

All of our challenges are challenges that can be mitigated to some extent, there are technical challenges that require meticulous planning and expertise in the field, but knowing our team these challenges will just become obstacles that we have to overcome and grow through.

# Opportunities

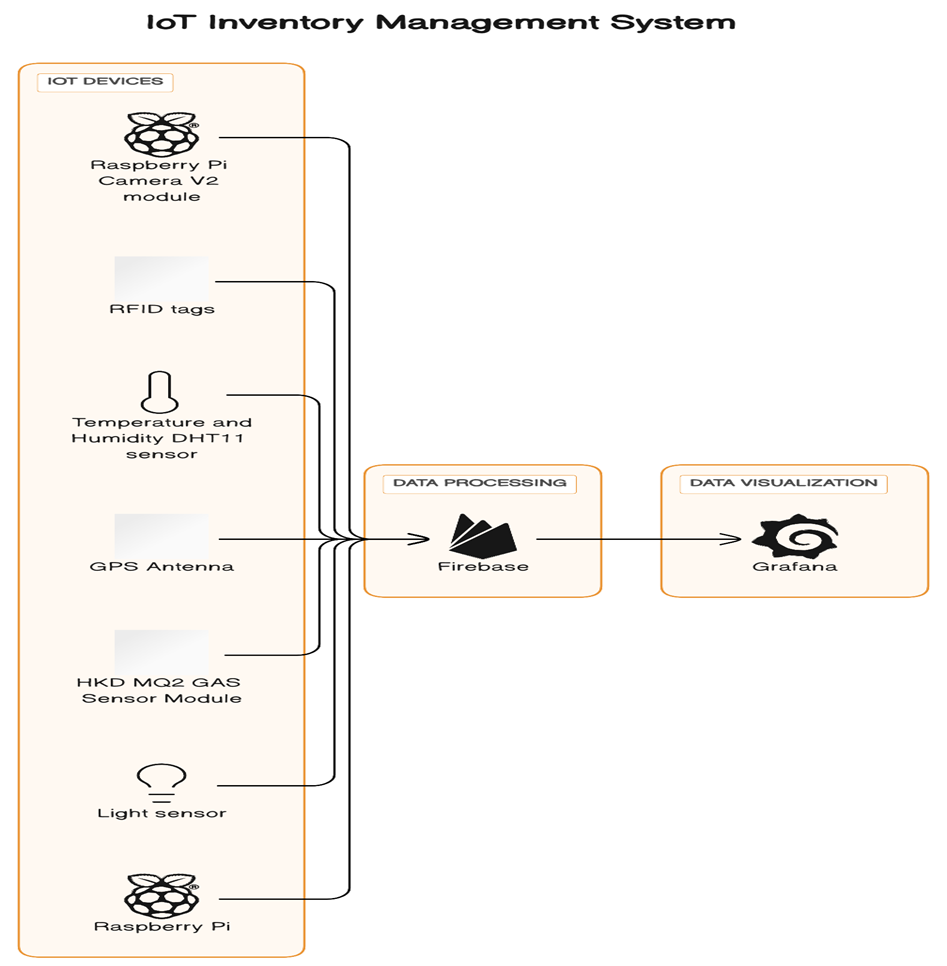
We predict many opportunities being brought up because of our solution and one of them is the reduction of theft of goods being imported/exported to and from South Africa. A GPS tracker will be attached to each container, making it easier for customers to locate where their order is in real time.

Ensuring product quality and safety is imperative in all industries. IoT solutions, with the ability to monitor and report on quality parameters, prove indispensable. Whether in trade or food production, the guarantee of quality is non-negotiable. IoT offers a versatile solution to this challenge by enabling comprehensive quality control.

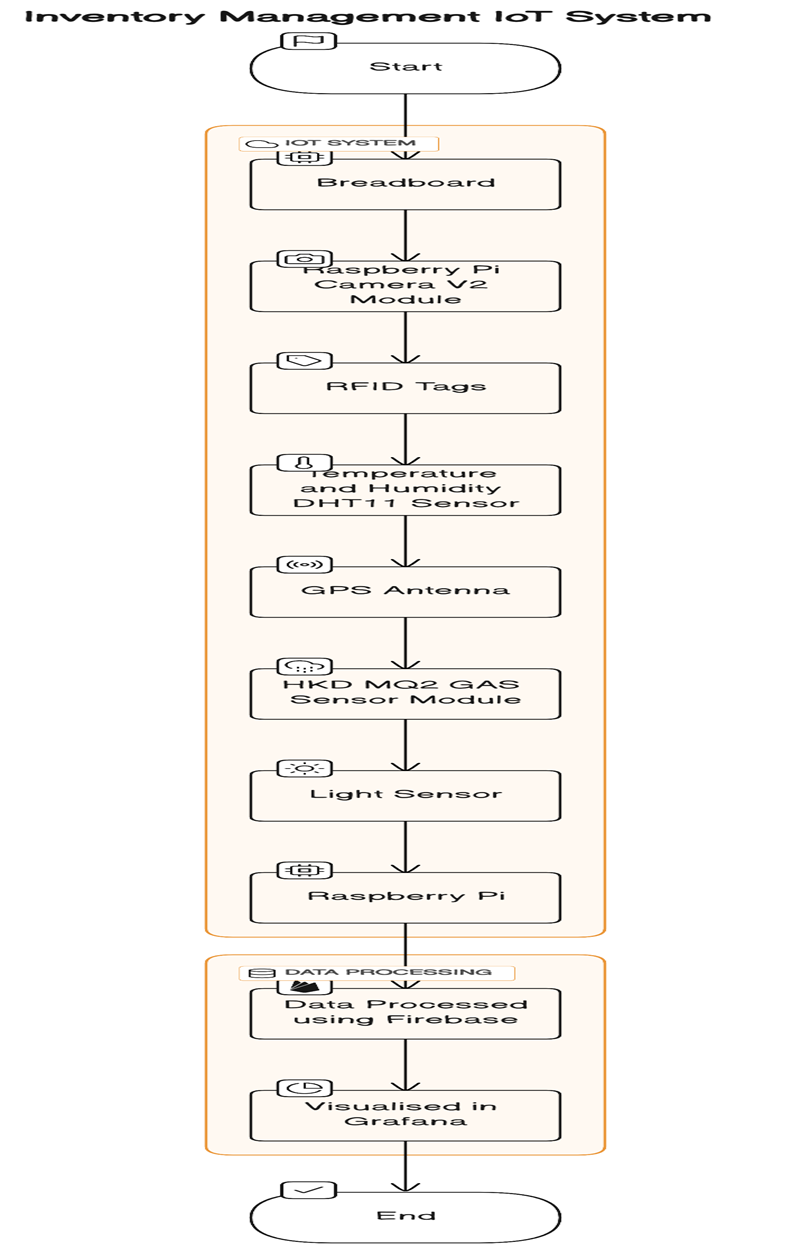
We could also implement the solution in the manufacturing and healthcare industries since effective inventory management is the backbone of a well-functioning industry. The solution is able to save costs and prevent operational disruptions. It ensures that products are readily available when needed and that resources are not wasted through overstocking.

Since South Africa is a developing country, we would like to make sure that customers around the world are satisfied with the quality of goods/services that we produce so that they may choose to invest in us and develop the nation.

# Solution architecture



# Data flow

****

The data flow in the inventory management systems starts with the Raspberry Pi as it serves as the main processing unit and connects to all the sensors. It collects data from the sensors, processes it, and communicates with the cloud or a central server for further analysis and storage.An RFID tag is the first sensor used to keep track of the quantity of goods using the different codes associated with the specific goods. RFID tags that can be attached to individual inventory items or containers. RFID readers connected to the Raspberry Pi can scan and track the movement of items in real-time, providing accurate asset tracking. The Raspberry PI camera V2 module is also incorporated into the system to capture images of the inventory, allowing for visual inspection and monitoring. Computer vision algorithms can be applied to analyse images for inventory tracking purposes. The DHT11 sensor detects sudden changes in temperature and humidity, providing valuable information about the condition and health of the perishables. GPS Antenna; GPS modules can be integrated into the IoT solution to track the location of inventory during transportation or when stored in different warehouses. This enables real-time tracking and monitoring of inventory movement. HKD MQ2 Gas Sensor module;Gas sensors can detect the presence of harmful gas or chemical in the inventory storage area. This helps in ensuring a safe environment and preventing damage to goods. The light sensor is then used for lighting and alerting purposes. The data collected from the sensors will be stored in real-time using Grafana and Firebase DB. Grafana is a popular open-source platform for data visualisation, while Firebase DB is a high-performance time-series database. These tools enable real-time monitoring and analysis of inventory data, providing insights into inventory levels, movement, and conditions.

# Code structure

## Temperature and humidity

**Import libraries:**

import Adafruit\_DHT

import firebase\_admin

from firebase\_admin import credentials, db

from datetime import datetime

import time

**Set sensor configuration:**

SENSOR\_TYPE = Adafruit\_DHT.DHT11

GPIO\_PIN = 4 # Replace with the actual GPIO pin number.

**Initialize firebase:**

CREDENTIALS\_PATH = "group3.json"

FIREBASE\_DB\_URL = 'https://iotgroup3-bafdb-default-rtdb.firebaseio.com/'

cred = credentials.Certificate(CREDENTIALS\_PATH)

firebase\_admin.initialize\_app(cred, {'databaseURL': FIREBASE\_DB\_URL})

**Setup firebase database reference:**

DATABASE\_REFERENCE = db.reference("tempandhum")

**Read sensor data in a loop:**

try:

while True:

humidity, temperature = Adafruit\_DHT.read\_retry(SENSOR\_TYPE, GPIO\_PIN)

**Send data to firebase:**

if humidity is not None and temperature is not None:

timestamp = datetime.fromtimestamp(int(time.time())).strftime("%m/%d/%y%H:%M:%S")

temperature = round(temperature, 2)

humidity = round(humidity, 2)

data\_to\_send = {

'timestamp': timestamp,

'temperature': temperature,

'humidity': humidity

}

DATABASE\_REFERENCE.child(timestamp).set(data\_to\_send)

print('Data sent to Firebase:')

print(f'Timestamp: {timestamp}')

print(f'Temperature: {temperature:.2f}°C')

print(f'Humidity: {humidity:.2f}%')

time.sleep(10) # Wait for 10 seconds before the next reading

**Handling reading failure and interruption:**

else:

print('Failed to retrieve data from the sensor. Check your wiring and configuration.')

except KeyboardInterrupt:

print('Script terminated by user.')

**Clean up resources:**

finally:

firebase\_admin.delete\_app(firebase\_admin.get\_app())

## Gas sensor

**Import libraries:**

import RPi.GPIO as GPIO

import time

**Set up GPIO mode and pin:**

GPIO.setmode(GPIO.BCM)

DO\_PIN = 7 # Replace with the actual GPIO pin number

GPIO.setup(DO\_PIN, GPIO.IN)

**Define Ethylene threshold:**

ETHYLENE\_THRESHOLD = GPIO.LOW # Replace with the actual threshold level for Ethylene gas

**Loop for gas detection:**

try:

while True:

gas\_present = GPIO.input(DO\_PIN)

if gas\_present == ETHYLENE\_THRESHOLD:

gas\_state = "Ethylene Gas Present"

else:

gas\_state = "No Ethylene Gas"

print(f"Gas State: {gas\_state}")

time.sleep(5.0) # Wait for a short period before reading again

except KeyboardInterrupt:

print("Gas detection stopped by user")

finally:

GPIO.cleanup()

## GPS sensor

Could not retrieve code.

## RFID

**Import libraries:**

import RPi.GPIO as GPIO

from mfrc522 import SimpleMFRC522

import csv

**Suppress warnings:and initialise reader:**

GPIO.setwarnings(False)

rfid = SimpleMFRC522()

read\_ids = set()

seen\_before = set()

**Set CSV file names and write headers:**

unique\_csv\_file = 'unique\_tags.csv'

duplicate\_csv\_file = 'duplicate\_tags.csv'

with open(unique\_csv\_file, mode='w', newline='') as unique\_csv:

unique\_writer = csv.writer(unique\_csv)

unique\_writer.writerow(['ID', 'Text'])

with open(duplicate\_csv\_file, mode='w', newline='') as duplicate\_csv:

duplicate\_writer = csv.writer(duplicate\_csv)

duplicate\_writer.writerow(['ID', 'Text'])

**Main loop for reading RFID tags:**

while True:

id, text = rfid.read()

if id not in read\_ids:

# New RFID Tag Detected

print("New RFID Tag Detected:")

print("ID:", id)

print("Text:", text)

# Write data to the unique CSV file

with open(unique\_csv\_file, mode='a', newline='') as unique\_csv:

unique\_writer = csv.writer(unique\_csv)

unique\_writer.writerow([id, text])

read\_ids.add(id)

seen\_before.discard(id) # Remove from seen\_before if it was there before

else:

if id not in seen\_before:

# Duplicate RFID Tag Detected. Ignoring.

print("Duplicate RFID Tag Detected. Ignoring.")

seen\_before.add(id)

## Camera and light sensor

**Import libraries:**

import cv2

import time

from picamera2 import Picamera2

from tflite\_support.task import core

from tflite\_support.task import processor

from tflite\_support.task import vision

import utils

**Set configuration:**

model = 'best.tflite'

num\_threads = 4

dispW = 1024

dispH = 768

**Initialize Picamera2:**

picam2 = Picamera2()

picam2.preview\_configuration.main.size = (dispW, dispH)

picam2.preview\_configuration.main.format = 'RGB888'

picam2.preview\_configuration.align()

picam2.configure("preview")

picam2.start()

**Configure object detection:**

pos = (20, 60)

font = cv2.FONT\_HERSHEY\_SIMPLEX

height = 1.5

weight = 3

myColor = (255, 0, 0)

fps = 0

base\_options = core.BaseOptions(file\_name=model, use\_coral=False, num\_threads=num\_threads)

detection\_options = processor.DetectionOptions(max\_results=4, score\_threshold=0.3)

options = vision.ObjectDetectorOptions(base\_options=base\_options, detection\_options=detection\_options)

detector = vision.ObjectDetector.create\_from\_options(options)

tStart = time.time()

while True:

im = picam2.capture\_array()

im = cv2.flip(im, -1)

imRGB = cv2.cvtColor(im, cv2.COLOR\_BGR2RGB)

imTensor = vision.TensorImage.create\_from\_array(imRGB)

detections = detector.detect(imTensor)

image = utils.visualize(im, detections)

cv2.putText(im, str(int(fps)) + ' FPS', pos, font, height, myColor, weight)

cv2.imshow('Camera', im)

if cv2.waitKey(1) == ord('q'):

break

tEnd = time.time()

loopTime = tEnd - tStart

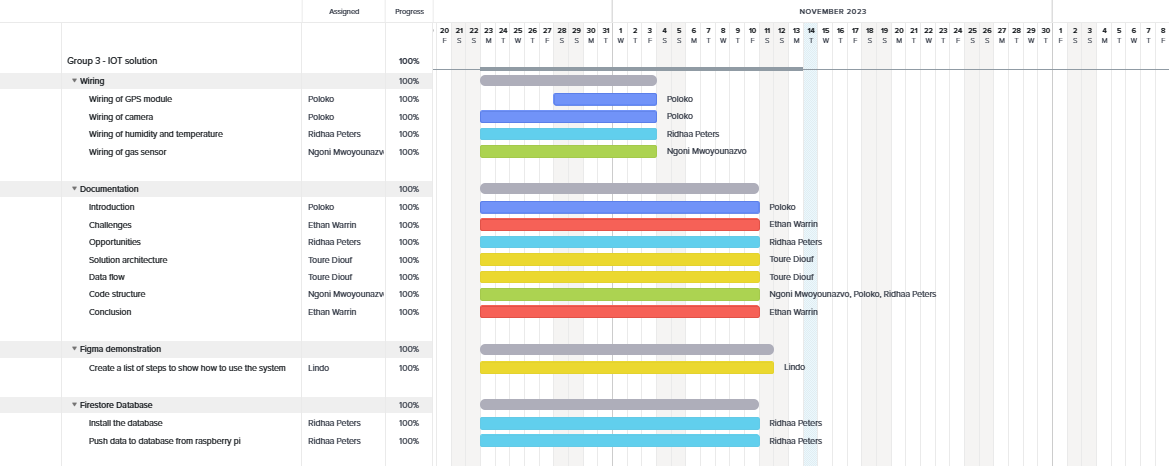
fps = 0.9 \* fps + 0.1 \* 1 / loopTime

tStart = time.time()

**Clean up:**

cv2.destroyAllWindows()

# Gantt chart



[Group 3 - IOT solution - Gantt | TeamGantt](https://app.teamgantt.com/projects/gantt?ids=3711181)

# What can be added/improved on?

* Weight sensor: Can be integrated to storage shelves, pallets, or containers to monitor the quantity of goods in real time . This will allow businesses to track inventory levels accurately .
* Pressure sensor**:** can be used to monitor the pressure levels within the storage container. This can help identify changes in pressure that may lead to damage or even impact the freshness.
* Vibration sensor: can detect excessive vibration during transportation or handling of perishable goods. This information can identify potential mishandling, allowing the businesses to take corrective actions.
* Device that cools the container if the temperature outside the container becomes too high.

# 

# Conclusion

In the fast-paced world of perishable goods logistics, efficiency and precision are paramount. With the successful development and implementation of our IoT solution, we have not only met these demands but have also set a new standard for the industry. Our innovative IoT system, seamlessly integrated with a user-friendly mobile application, has revolutionised the way perishable goods are managed and transported from the source at farms, to the consumer's table. This achievement underscores the power of technology in optimising every facet of the supply chain, which is our greater goal.

Our IoT solution's contribution extends far beyond logistics, influencing the realms of trade, agriculture, and consumer satisfaction. By offering real-time monitoring, tracking, and quality control, we've introduced a heightened level of assurance in the quality and safety of perishable goods. The intuitive user interface (UI) and user experience (UX) of our mobile application have put the power of information directly into the hands of all stakeholders. With our IoT solution, producers can now ensure the quality of their goods, logistics managers can optimise routes and inventory, and consumers can trace the journey of their food with ease. This successful creation of our IOT system shows the potential of technology to reshape industries, proving that our IoT solution has not only arrived but is poised to thrive in an ever-evolving global market.

# Delegation of tasks

| **Name and Surname** | **Task(s) completed** |
| --- | --- |
| Ridhaa Peters | * Installation of firebase and Grafana * Setting up the temperature and humidity sensors. * Pushing data from temperature and humidity sensor to firebase in realtime. * Pushing temperature and humidity as well as gas sensor data to grafana in real time. * Adding the “opportunities” in the documentation. * Managing Trello and TeamGantt * Code structure |
| Toure Nogaye Diouf | * Documentation * Gas sensor * Powerpoint presentation |
| Ngoni Mwoyounazo | * RFID tag, GPS module, Gas sensor * Installation of libraries * Documentation * Website * Debugging |
| Poloko Khoana | * Raspberry PI camera, Light sensor * Installation of libraries * Structuring the IoT system * Debugging codes * Handling of infrastructure * Documentation |
| Mapitsi Tsebe | * Documentation * Gas sensor |
| Ethan Edward Warrin | * Documentation * Light sensor |
| Lindokuhle Zophila Nzama | * Prototype |

# 

# 

# 

# 

# 

# Reference List

Kriege, M. (2023, October 5). Declining demand for perishable goods: How must supply chains adapt ?. Maersk. <https://www.maersk.com/insights/resilience/2023/10/05/demand-for-perishable-products#:~:text=Supply%20chains%20handling%20temperature%2Dsensitive,resulting%20in%20significant%20financial%20losses>

Mecalux. (n.d.). *Perishable goods: Storage systems and examples in Logistics*. Interlake Mecalux, Warehouse solutions. <https://www.mecalux.com/blog/perishable-goods#:~:text=Logistics%20characteristics%20of%20perishable%20goods,-When%20it%20comes&text=To%20stave%20off%20the%20proliferation,the%20final%20product%20to%20deteriorate>

Microsoft. (n.d.). *What is the internet of things (IOT)?*. What Is the Internet of Things (IoT)? | Oracle South Africa. <https://www.oracle.com/za/internet-of-things/what-is-iot/#:~:text=The%20Internet%20of%20Things%20(IoT)%20describes%20the%20network%20of%20physical,and%20systems%20over%20the%20internet>

Nguyen, A., & Nguyen, A. (2023, June 30). *Outsourcing to South Africa: Benefits, challenges, industries*. Time Doctor Blog. <https://www.timedoctor.com/blog/outsourcing-to-south-africa/>