Initialisation

* Power up the raspberry Pi
* Identify the ip address of the raspberry Pi, 192.168.0.36
* Navigate to the 192.168.0.36:1880 , Distributed Storage System - Pi
* Power up the weight scale
* Open up the command prompt, to start Node-RED, run node-red
* Navigate to the local host:1880, Distributed Storage System – Client
* Connect the weight scale to the raspberry Pi using the USB cable.
* Refresh the dashboard at the client end.
* Connect the raspberry Pi via SSH using the putty
* Open the putty console
* Run the command, ls -l /dev/tty\* or Run the command, dmesg | grep "tty" which tells the serial port where the USB cable is connected.
* Specify this serial port while configuring the serial.in node (scale) in the Node-RED.

Select Article from dropdown.

Server End Flow

Use-cases

* Monitor the weight
* Monitor the connection

Monitor the weight includes --🡪 Monitor the connection

Activity diagram

Monitor the weight

* Place some object on the weight scale
* Parse the raw weight data from the weight scale
* Format the weight information.
* Display the weight on the dashboard.
* Send the weight to the local machine via MQTT on the topic ‘inv’.

Monitor the connection

* Check if the weight scale is connected or not by checking the status message of the serial.in node.
* If Connected
  + Connection string
  + Extract the serial number of the raspberry Pi by running the script piserial.sh
  + Extract the device number of the device connected to raspberry Pi by running the script devicenum.sh
  + Combine all the above info together as a unique ID for the weight scale.
  + Send this unique ID and connection status to the client via MQTT on the topic ‘scale\_id’.
* If not connected
  + Send the connection status to the client via MQTT on the topic ‘scale\_id’.

Client End Flow

Use-case diagram

Three broad use case categories

* Display Inventory
  + Weight scale connection status
  + Article Selection
  + Display selected article Inventory

Display selected article Inventory includes ------> Article selection

* Modify (DB management)
  + Add new article
  + Delete an article
  + Log into DB
  + Delete log
  + Download log

Log into DB 🡨--extends Delete Log

Log into DB 🡨-- extends Download Log

* Display Article match/mismatch
  + Display method 1 (error.py)
  + Display method 2
  + KNN method
  + Decision tree method

KNN method includes ------> Download Log

Decision tree ------> Download Log

Activity Diagram

Weight scale connection status

* MQTT subscriber subscribes on the topic ‘scale\_id’
* Parse the info into connection status and unique scale.
* Display the connection status
* Display the unique scale id.

Article Selection-

* Check if any article name is present or not in dropdown
* If present,
  + Get the article number and unit weight using SELECT SQL query in distributedstorage----(1)
  + Display the article number and the unit weight.
* If not present,
  + Add new article
    - Specify article name
    - Specify article number
    - Scan the unit weight.
    - Run the Insert SQL Query

Display selected article inventory

* MQTT subscriber subscribes on the topic ‘inv’ to get the current weight information.
* Display the current weight info.------------------------------(3)
* (1)---Compute the number of pieces
* Display the pieces information.--------------------------------(4)

Modify

Delete article

* Specify the article to be deleted from the dropdown.
* Prompt to confirm the deletion of the specified article.
  + If OK,
    - Run the DELETE query in distributedstorage

Log into DB

* Toggle the slide bar ON
* Prompt to confirm the logging activity
  + If OK,
    - Run the INSERT query into inventorylog

Delete Log

* Toggle the slide bar ON
* Prompt to confirm the deletion of the Log
  + If OK,
    - Run the DELETE query in inventorylog.

Download Log

* Toggle the slide bar ON
* Run the SELECT \* query in inventorylog to extract all the data.
* Convert into csv
* Store as log.csv ---------------------------------------------(2)

Article match/mismatch

Method 1

* (1)----------------
* run the error.py to compute the array containing the permissible weight values.
* Compare the (3) and check if it is present in the permissible weight values.
* If present,
  + Display ‘article matches’
* If not,
  + Display ‘suspects article mismatch’.

Method 2

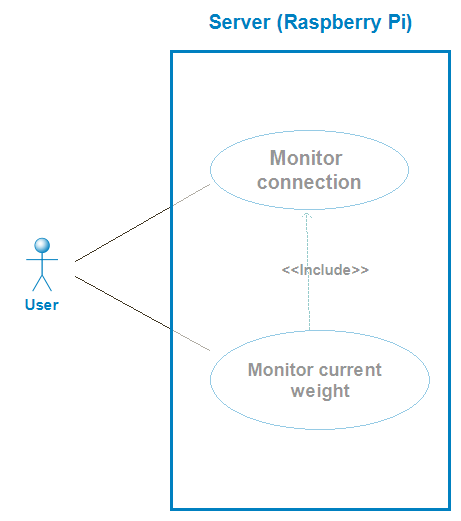
* (3),(1),(4)
* Calculate the unit weight \* pieces
* Compare with the obtained weight with tolerance of 4
* If true,
  + Display ‘article matches’
* If false,
  + Display ‘Kindly check the weight scale’.
  + Generate an alert notification.

KNN method

* (3)
* (2)
* Run the knn\_1f.py to predict the class label (article number)
* Display the predicted article number

Decision tree method

* (3)
* (2)
* Run the dtc\_1f.py to predict the class label (article number)
* Display the predicted article number



(