SMARKT - Smart Supermarket

Programming for IOT applications(01QWRBH) – Final Project

Daniel Gaiki (s232546)

ICT4SS
Information and Communications Technologies for Smart Societies

SMARKT – General System

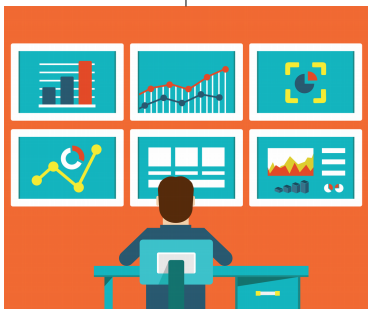
Electronic Trolley:

Raspberry Pi + LCD interface +
weight sensor + barcode Scanner

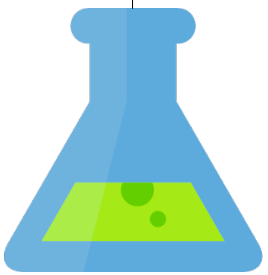
Firebase

Payment and Checkout control:

Raspberry Pi + LCD interface +
Keyboard + barcode Scanner

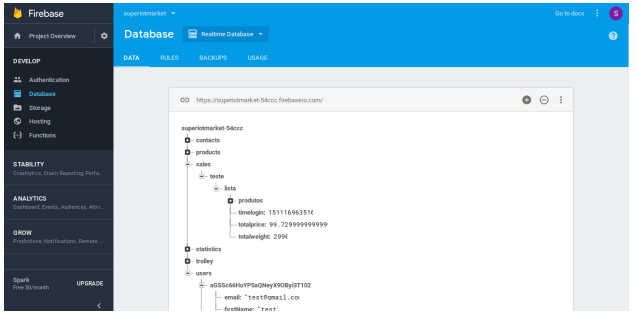


Dashboard- Freeboard



Prediction of future stock or
consumption

SMARKT – FIREBASE



Control.py

- Time of scanning
- Scanned Prod.
- Total weight
- Time weight meas.
- Trolley #

HTTP post/get

payment_check.py

HTTP post/get

- Payment verification.
- Time of payment
- User
- Mode of payment

Firebase

Display HTML + AngularJS

- Login
- Register new user
- View shop list
- Confirm shop list

HTTP post/get

dashboard prediction.py

statistics about purchases

generate a image

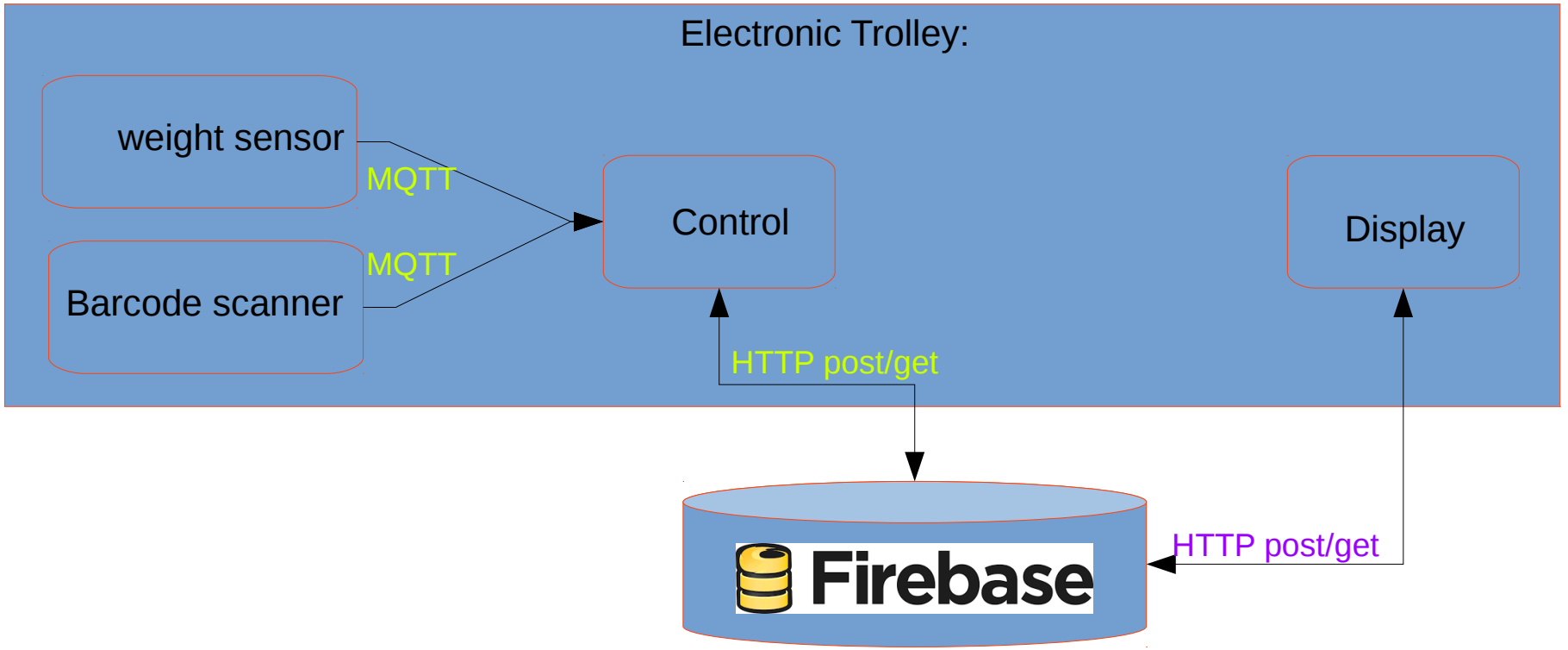
HTTP post/get

calc_listas.py

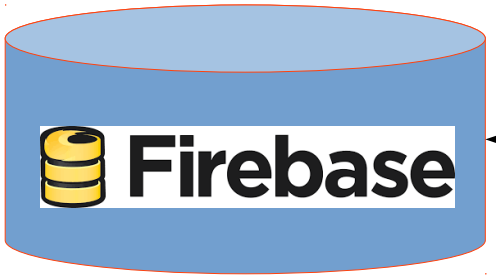
- Total Price
- Total weight
- Create historic info user- Statistics

ICT4SS
Information and Communications Technologies for Smart Societies

SMARKT – Electronic Trolley



SMARKT – FreeBoard



HTTP post/get

Dashboard FREEBOARD

- Statistics about quantity of purchased products
- Graphic generation
- Prediction of future consumption

SMARKT – Payment and Checkout

People passage control

- Open close
- Light indicator of free passage
- Alarm

MQTT

Payment and checkout

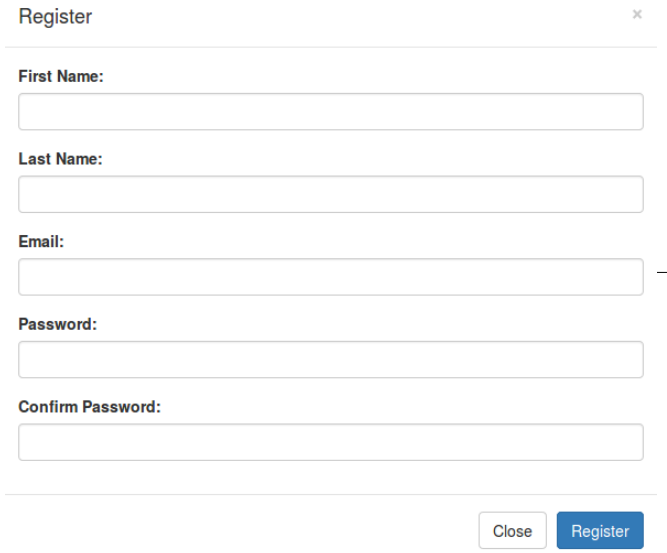
- Payment OK
- Send message to passage control
- Scan trolley
- Send mode of payment
- Send time of payment

Firebase

HTTP post/get

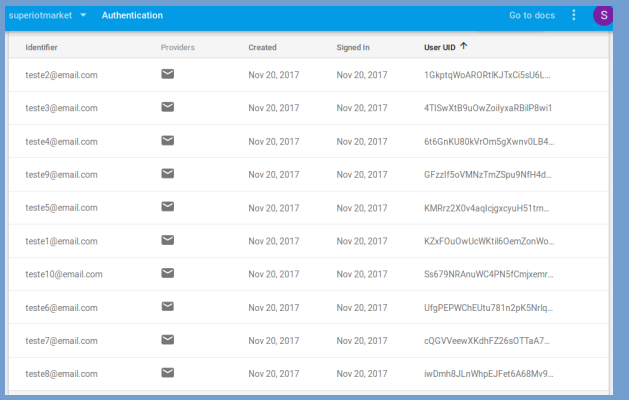
SMARKT – Trolley shop interface

1- REGISTER NEW USER



HTTP/POST

Firebase



SMARKT – Trolley shop interface

2- LOGIN

HTTP/POST

SMARKT – Scanner Barcode

3- Scan Barcode

import cv2.cv as cv #Use OpenCV-2.4.3
import zbar

MQTT

Control.py

Product exist in the firebase list?

N

discard

y

Product exist in the User list?

N

y

HTTP

HTTP

FIREBASE

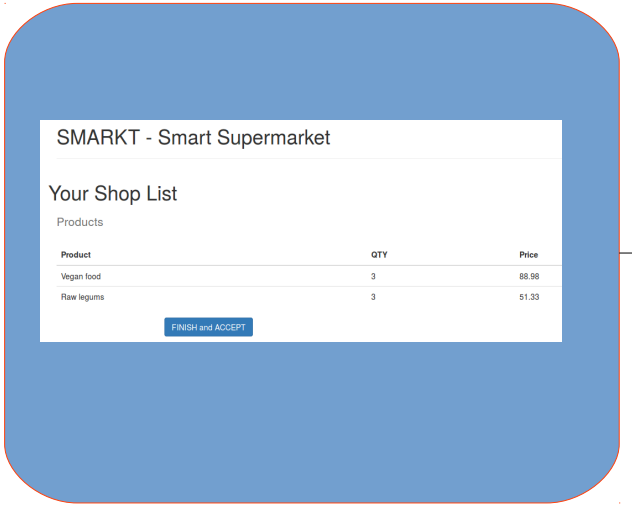
Increase QTY

ADD on user shop list

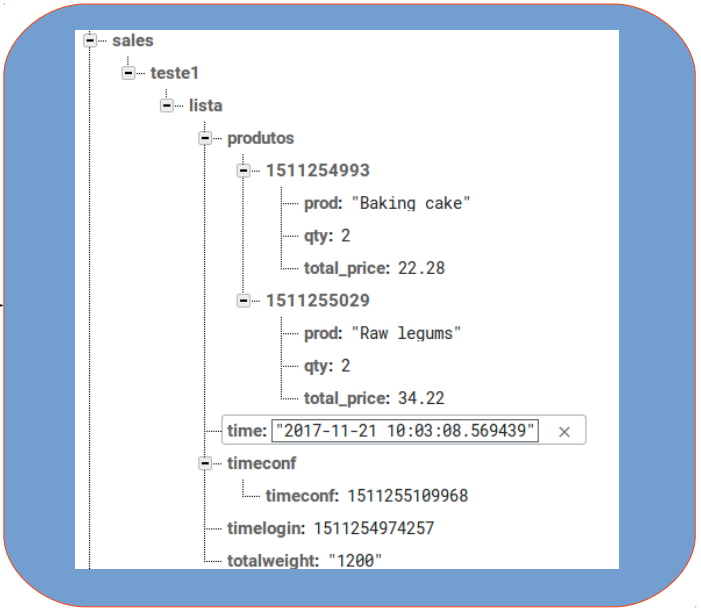
SMARKT – Final Shop accept list

3- Verify final shop list

Shoplist confirmation



Generation of timeconf on firebase



SMARKT – Final Calculation on shop list

4- Final shop list calculation on FIREBASE

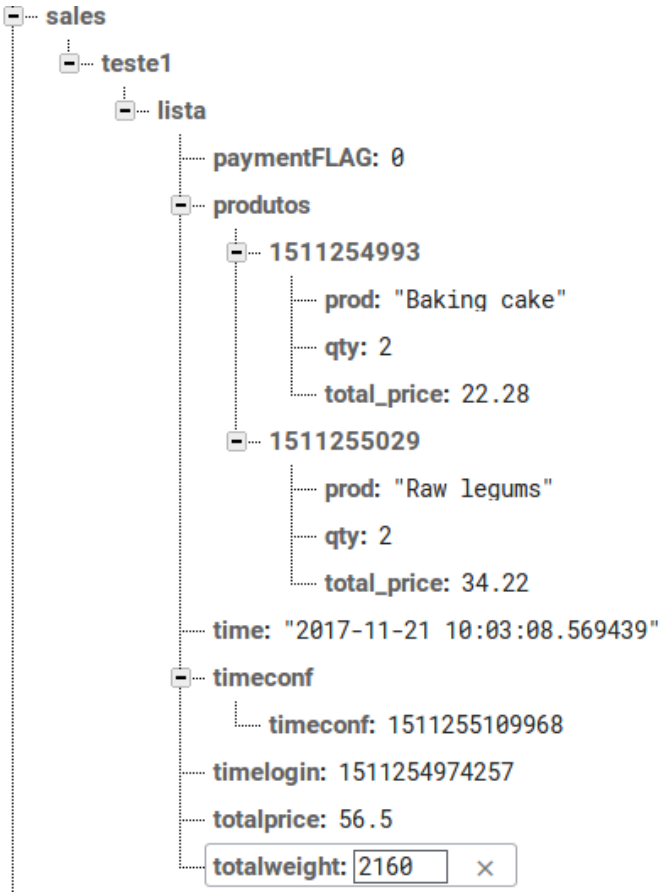
Generation of:

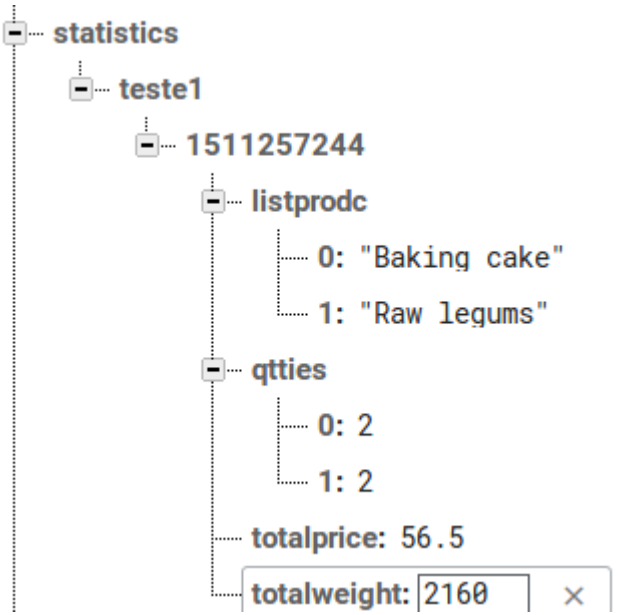
Total Price

Total Weight

Payment FLAG

Statistics





SMARKT – Entire system codes

Interface

Index.html

views	js	controllers
add.html	app.js	addcontroller.js
edit.thml	script.js	editcontroller.js
list.html	security.js	listcontroller.js

Python

Client
- Scanner.py
- Control.py
- weight.py

Firebase management
- calc_listas.py

Supermarket payment control
- payment_check.py

dashboard
- prediction.py

SMARKT – Entire system codes

Interface explanation

Index.html General Vision of the screen, Register and Login screen

add.html Vision of purchased items by the user.

addcontroller.js Javascript connection between Firebase and add.html code, responsible to get purchased items on the user list on firebase.

Python explanation

Client

Scanner.py Using OpenCV library, get image from webcam and remove from a barcode image the code, send the message by MQTT to control.py
Message Topic:j_temp = '{"id":"trolley#","sensor": "scanner","reading":'"+str(codigo)+'", "time":'"+hora_s+'"}'

weight.py Measure the weight of total amount of purchased products of the client, senf message to control.py

Message Topic: j_temp = '{"id":"trolley#","sensor": "weight","reading":'"+str(ins)+'", "time":'"+hora_s+'"}'

Control.py Receive message from Scanner and weight sensor and send to Firebase database, control the login of user on the trolley, remember, the trolley has a fixed number.

SMARKT – Entire system codes

Python_explanation

Supermarket payment control	payment_check.py	Scan the trolley, verify if the weight measured by trolley sensor has the same value of the database value, if the user pay the purchase , send a message to control passage authorizing the passage to the user.
Firebase management	calc_listas.py	Control all information in the firebase, calc the total amount of each user, and the total weight of each purchase.
dashboard	prediction.py	Verify the statistics about purchases of the products on the history, quantity of each product buyied in specific week day, play a regression and predict the future purchase, generate a image and sent to the firebase.

SMARKT - Smart Supermarket

We are hiring too!

:)

Thanks for watching!

ICT4SS
Information and Communications Technologies for Smart Societies