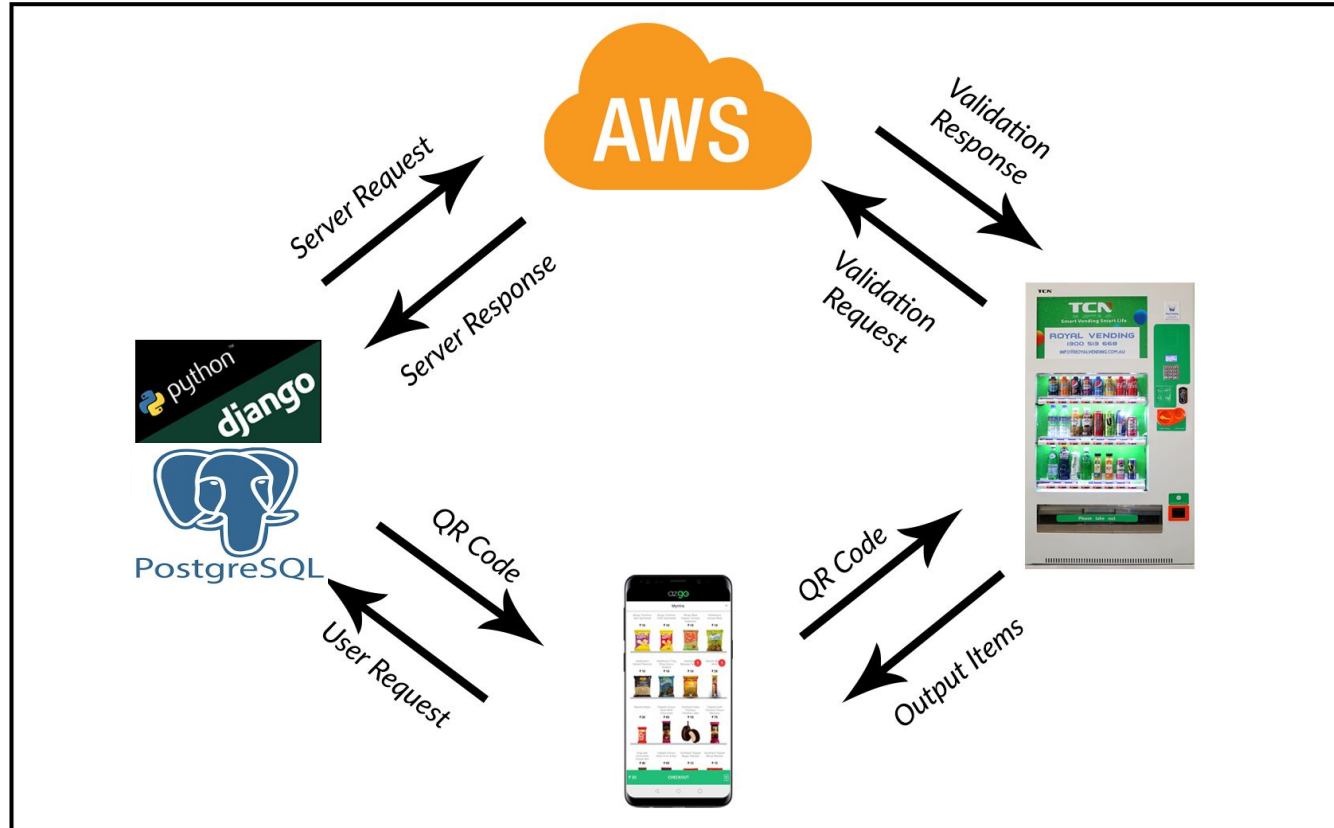


Smart Vending Machine

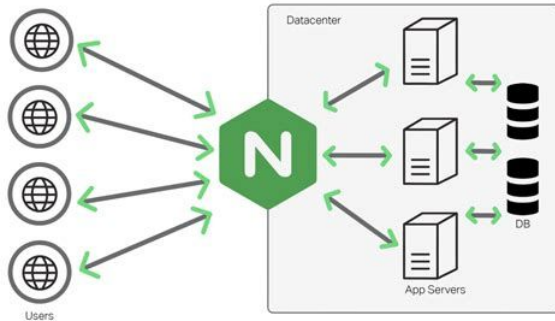
Milestone 4

Solution Overview



Till Now..

- Front End
- Back End
- Rest API
- Hardware Design



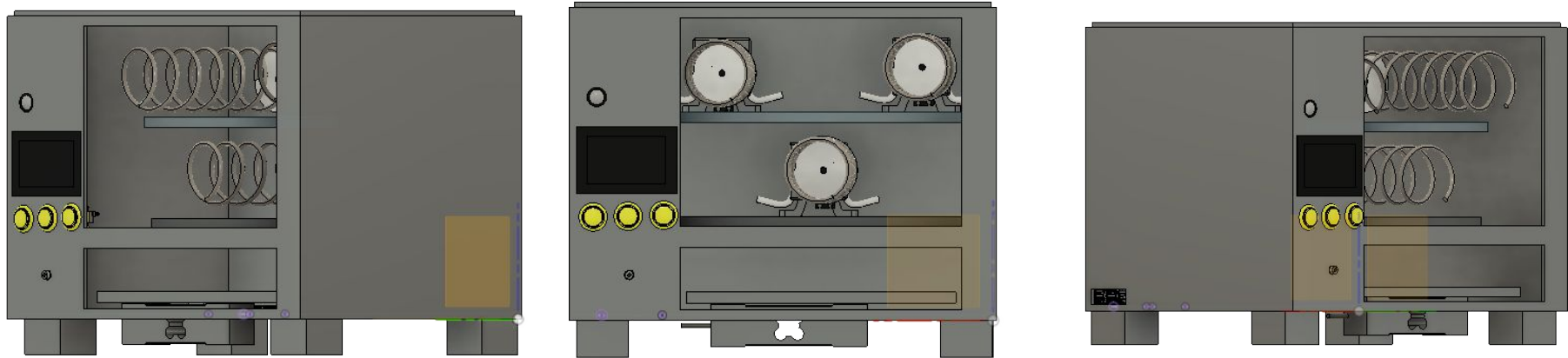
Visit our website



<http://ec2-54-87-72-111.compute-1.amazonaws.com>

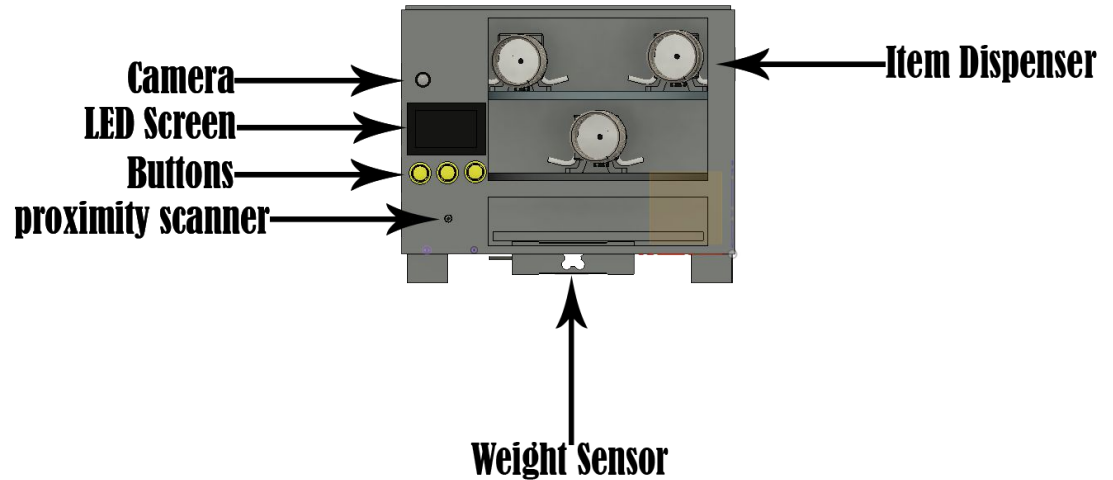
<https://smart-happyvending.herokuapp.com>

Overall Design



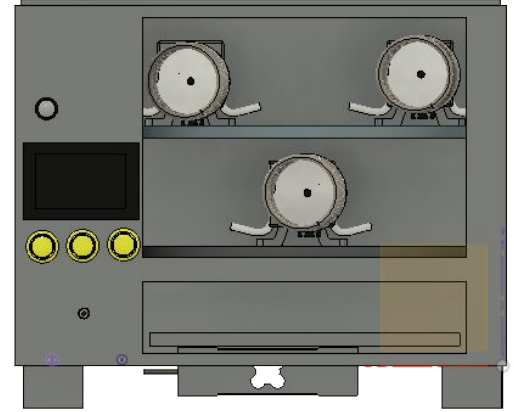
Device Design

- Full Closed Structure
- Rigid Wood body
- PIR controlled Power distribution
- LED Display



User Experience

- LED Display to interact
- Easy Set up Just have to interact with phone
- 24hrs service
- Easy Cancellation at any time
- LED strips inside to make sure the items are visible

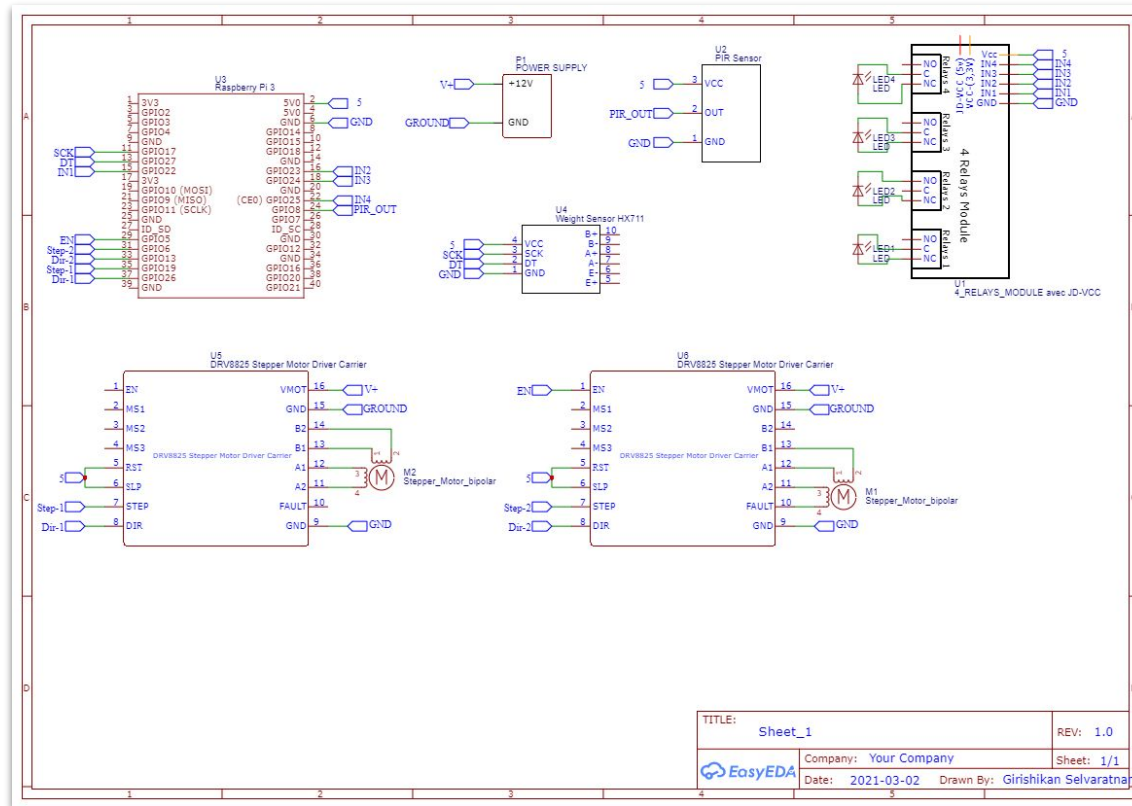


Hardware design Aspects

- **Scalability**
 - Multiple Devices Single server
 -
- **Reliability**
 - Weight Sensor to make sure the dispensing of an item
 - Stepper motor to make sure the Quantity of the item
- **Security**
 - 6 pin number security to the door of the vending machine
 - Rigid wooden body(demonstration)/real vending machine is metal body



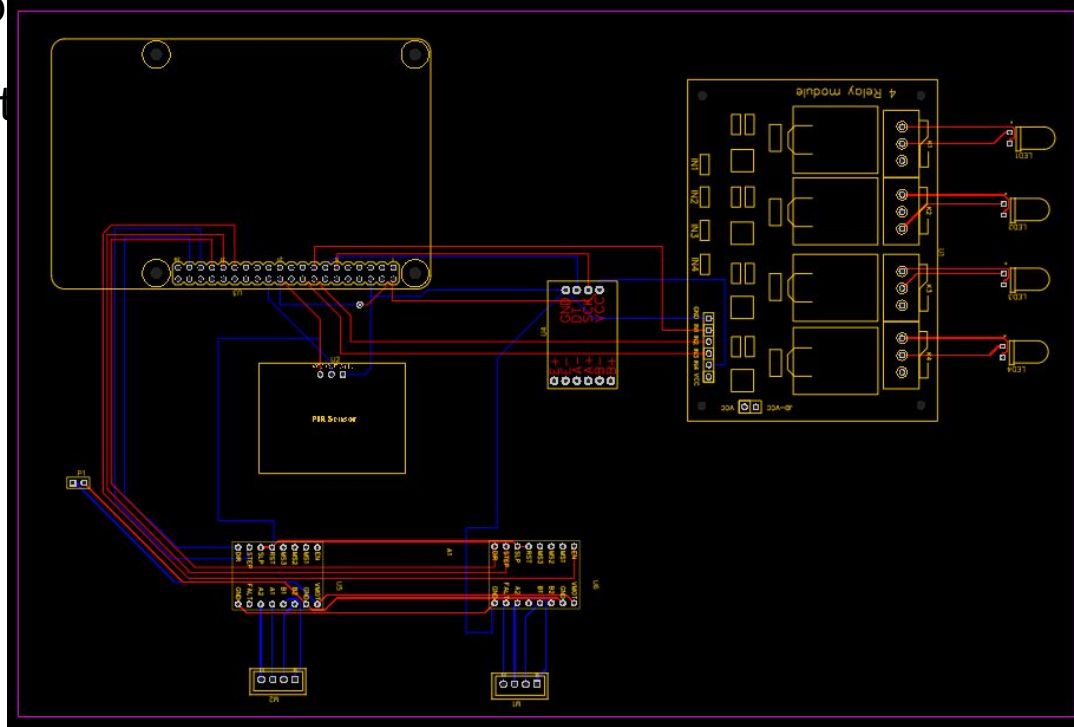
HardWare Design



PCB Design

Double Side printing

Use PCB Print



Hardware Testing

What we have To Test?

- Every Hardware Components
- Raspberry Pi 3 and Components

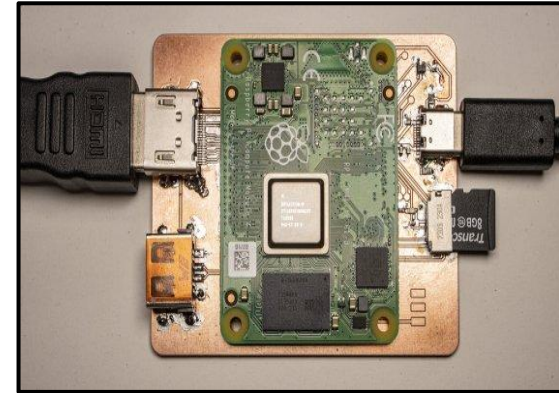
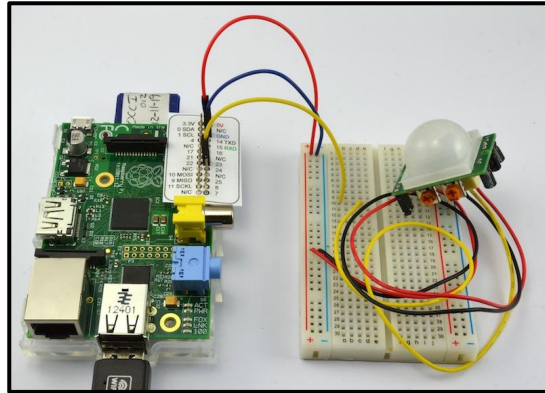
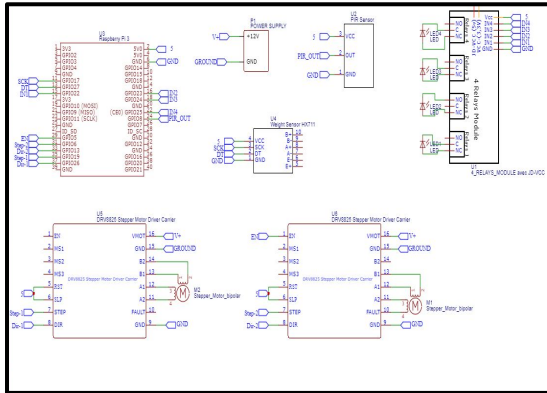
Why this Hardware testing is needed?

- High Risks involved
- Product will loose the market



Hardware Testing

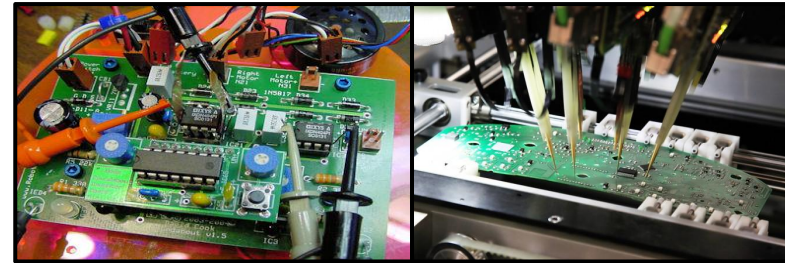
Before Testing We have to check the circuit correctly working or not by with breadboards.



Hardware Testing

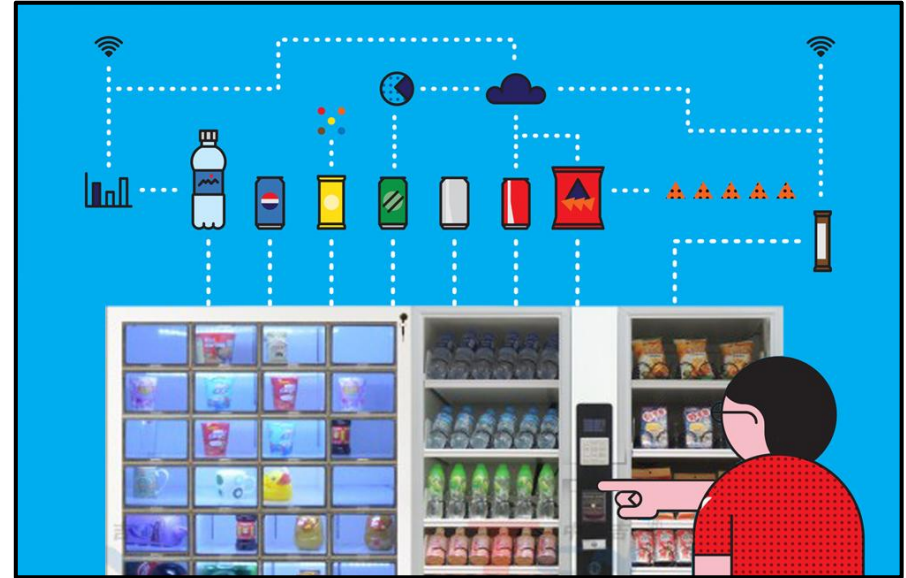
What we have To Test?

- Board Testing
- ❖ Why Need this testing?
 - Board got burned due to power failure
 - Require voltages are not coming
- ❖ Testing Methods
 - Check the power sections using Multimeter
 - Check the Connectivity by Blinking LED



Hardware Testing

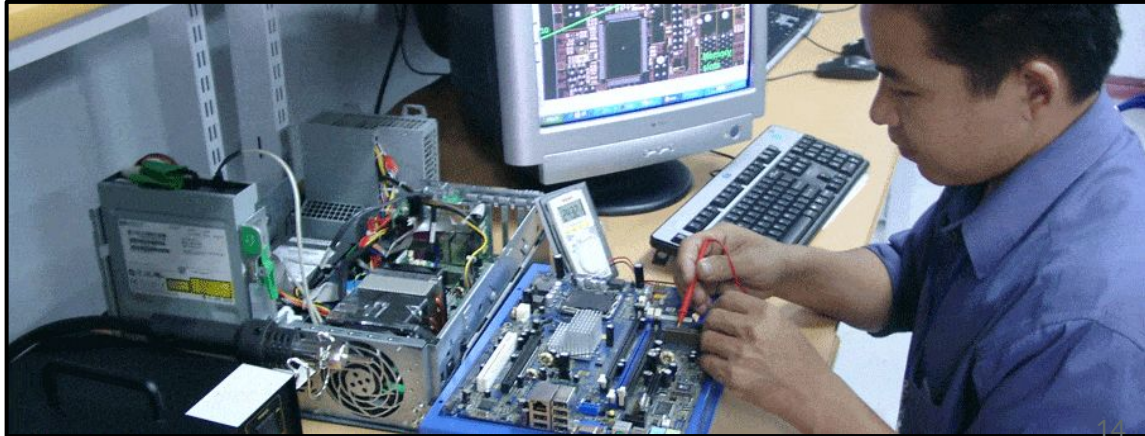
- **Performance Testing**
 - ❖ Server side
 - ❖ Client Side
- **Embedded System Testing**
 - ❖ System Unit Testing
 - ❖ System Integration Testing
 - ❖ System Validation Testing



Testing Challenges

Difficulties we may Face through this Testing:-

- Hardware Dependency
- Software and hardware defects
- Reproducible Defects
- Continuous Software updates
- Defect Retest Difficulties
- Open Source Software



Software Aspects

- **Scalability**
 - Multiple user access to the server
 - Pagination for the products
- **Reliability**
 - Email Authentication
 - Reset forgot Password
 - Paypal Payments



Rest API

JWT(json web token) authentication

CURD operations available

With Rate limit

Django REST framework

Wrapper Func

Wrapper Func

GET ▾

GET /api/orders/

HTTP 401 Unauthorized

Allow: OPTIONS, GET, POST

Content-Type: application/json

Vary: Accept

WWW-Authenticate: Bearer realm="api"

```
{
  "detail": "Authentication credentials were not provided."
}
```


QR Code and Scanning Process

```
karikaranvetti@linux: ~/5th_semi/final_deployed_version/happyvending-live
karikaranvetti@linux:~/5th_semi/final_deployed_version/happyvending-live$ python3 machine.py
hey !! giri Your Order is Approved
You can Get : {'Snickers': 3, 'Coca-Cola': 1, 'Red Bull': 1}
karikaranvetti@linux:~/5th_semi/final_deployed_version/happyvending-live$
```

```
machine.py
4
5 from PIL import Image
6 from requests.exceptions import HTTPError
7
8 def validate(qrData, serverData):
9     valid=False
10     serverItems={}
11     for item in serverData['OrderItems']:
12         serverItems[item['name']]=item['quantity']
13         if((qrData['transactionComplete']==serverData['complete'])
14             and (qrData['transactionstatus']==serverData['status']))
15             and (float(qrData['transaction_id'])==float(serverData['transaction_id']))
16             and (qrData['transactionstatus']==serverData['status'])
17             and (int(qrData['amountToday'])==int(serverData['paidAmount']))
18             and (serverItems==qrData['orderProducts'])
19             and (qrData['customerName']==serverData['customer']):
20                 valid=True
21     return valid
22
23 #qr code reading
24
25 d = decode(Image.open('qr_code.png'))
26 qrData=json.loads(d[0].data.decode('utf-8'))
27 # print(qrData)
28
29 #server request
30 load={
31     "username": "karikaran",
32     "password": "Test1357"
33 }
34
35 try:
36     response = requests.post('http://ec2-54-87-72-111.compute-1.amazonaws.com/api/api/token/', json=load)
37     data=ast.literal_eval(response.text)
38     response.raise_for_status()
39
40 except HTTPError as http_err:
41     print(f'HTTP error occurred: {http_err}') # Python 3.6
42 except Exception as err:
43     print(f'Other error occurred: {err}') # Python 3.6
44
45 # print(type(response.text))
46
47 header={}
48 header['Authorization']='Bearer '+ data['access']
49 # print(header)
50 url='http://ec2-54-87-72-111.compute-1.amazonaws.com/api/orders/'+qrData['id']+'/'
51
52 try:
53     response = requests.get(url,headers=header ,)
54     response.raise_for_status()
55
56 # data1 = response.read()
57 jsonData = json.loads(response.text)
58 # print(jsonData)
59 serverData=jsonData
60
61 Valid =validate(qrData,serverData)
62
63
64
65
66
67
68
69
70
71
72
73
```

Demonstration Plan

Web Application to Browse the products

To check the Payment systems we have created a Paypal sandbox

A sample Vending machine to interact with

3 different Items to dispense

And refilling the items



Q & A