



**Dharmsinh Desai University, Nadiad**

**Faculty of Technology**

**Department of Computer Engineering**

**B. Tech. CE Semester – IV**

**Subject: SP+SEPP**

**Project title: E-Waste Management System**

By:

1) DABHI JAYESH BHARATBHAI CE022 19CEUBS076

2) DHARAVIYA ROHIT DAYALAJIBHAI CE025 19CEUBG040

## E-WASTE MANAGEMENT SYSTEM

E-waste or Waste Electrical and Electronic Equipment (WEEE) are loosely discarded, surplus, obsolete, broken, electrical or electronic devices. ... Improper dismantling and processing of e-waste render it perilous to human health and our ecosystem. Therefore, the need of proper e-waste management has been realized.

India ranks 177 amongst 180 countries and is amongst the bottom five countries on the Environmental Performance Index 2018, as per a report released at the World Economic Forum 2018. This was linked to poor performance in the environment health policy and deaths due to air pollution categories. Also, India is ranked fifth in the world amongst top e-waste producing countries after the USA, China, Japan, and Germany and recycles less than 2 per cent of the total e-waste it produces annually formally. Since 2018, India generates more than two million tonnes of e-waste annually, and also imports huge amounts of e-waste from other countries around the world. Dumping in open dumpsites is a common sight which gives rise to issues such as groundwater contamination, poor health, and more. The Associated Chambers of Commerce and Industry of India (ASSOCHAM) and KPMG study, Electronic Waste Management in India identified that computer equipment account for almost 70 per cent of e-waste, followed by telecommunication equipment phones (12 per cent), electrical equipment (8 per cent), and medical equipment (7 per cent) with remaining from household e-waste.

# INDEX

- 1 Abstract
- 2 Introduction
  - 2.1 Project Details: Brief Introduction
  - 2.2 Technology and Tools Used
- 3 Software Requirement Specification
- 4 Design.....
  - 4.1 Use Case Diagram
  - 4.2 Sequence Diagram
  - 4.3 Activity Diagram
  - 4.4 Data Flow Diagram
  - 4.5 Class Diagram
- 5 Implementation Details
- 6 Screen-shots of the System
- 7 Limitation and Learning from project
- 8 Bibliography

# APPLICATION

## How much e-waste are we generating and why should we worry about it?

According to a [2019 United Nations report](#), titled ‘A New Circular Vision For Electronics, Time for a Global Reboot’ consumers discard 44 million tonnes worth of electronics each year; only 20 percent is recycled sustainably.

The Global E-Waste [Monitor](#) 2020 shows that consumers discarded 53.6 million tonnes worth of electronics in 2019 globally, up 20 percent in 5 years.

India [generated](#) 3.2 million tonnes of e-waste last year, ranking third after China (10.1 million tonnes) and the United States (6.9 million tonnes). Following the current growth rate of e-waste, an ASSOCHAM-EY joint [report](#), titled ‘Electronic Waste Management in India’ estimated India to generate 5 million tonnes by 2021. The study also identified computer equipment and mobile phones as the principal waste generators in India.

With COVID-19 keeping people indoors, the usage is only getting higher; and without proper intervention, it is likely to be over 100 million tonnes by 2050.

## What Can be Done to Reverse This Trend?

### **Equipment Reuse**

The [U.S. Environmental Protection Agency \(EPA\)](#) published a policy in June 2002 in the Federal Register that noted, “because obsolete electronics are often capable of reuse, they are not considered wastes until a decision is made that they cannot or will not be reused.” Qualified individuals, such as resellers or recyclers, usually make this decision. This guidance allows managers to collect unwanted electronics without regard to waste rules.

Computers and other electronic equipment less than five years old generally qualify for reuse. A manager's initial step is to determine if one department's equipment might benefit other departments within the organization.

The second option is to donate used electronics to a refurbisher or reseller of used equipment, first making sure to understand the types of products the refurbishing firm accepts. Managers also need to remember to include accessories, such as cords, cables, modems, the mouse, keyboards, speakers, software, printers and ink cartridges.

These companies can ensure the equipment is operational and handle recycling or disposal of waste materials refurbishing might generate. An electronics-refurbishing firm also can make proper upgrades to optimize the equipment for use.

The last option for managers is to donate electronic equipment directly to a not-for-profit charity or school.

## Technologies/tools used :

1. Python 3.9
2. Django 3.1.7
3. Vs code
4. HTML, CSS, JAVASCRIPT
5. Bootstrap
6. JQuery
7. MySQL

# Software Requirement Specification

## 1.Manage User:

### R.1.1: Login Account

- Description:user can login account
- Input:enter username and password
- Output:Message("Successfully logged in")

### R.1.2: Create Account

- Description:user can create account
- Input:enter username and password and email
- Output:account created and logged in

### R.1.3: Logout Account

- Description:user can logout account
- Input:click on logout
- Output:user is logged out

## 2.Manage Cart:

### R.2.1: Add to cart

- Description:user can add product to cart
- Input:click on add to cart
- Output:Message("product added to cart"). Cart shows product

### R.2.2: Remove from cart

- Description:user can remove product from cart
- Input:click on X button
- Output:product is removed from cart

### 3.Manage Wishlist:

#### R.3.1: Add to Wishlist

- Description:user can add product to wishlist
- Input:click on heart button
- Output:product added to wishlist

#### R.3.2: Show wishlist

- Description:user can show wishlist
- Input:click on wishlist.
- Output:wishlist is shown.

### 4.Place order:

#### R.4.1: Place order

- Description:after adding product to cart user can checkout
- Input:click on checkout and fill details and click on place order
- Output:order is placed

### 5.Seller account:

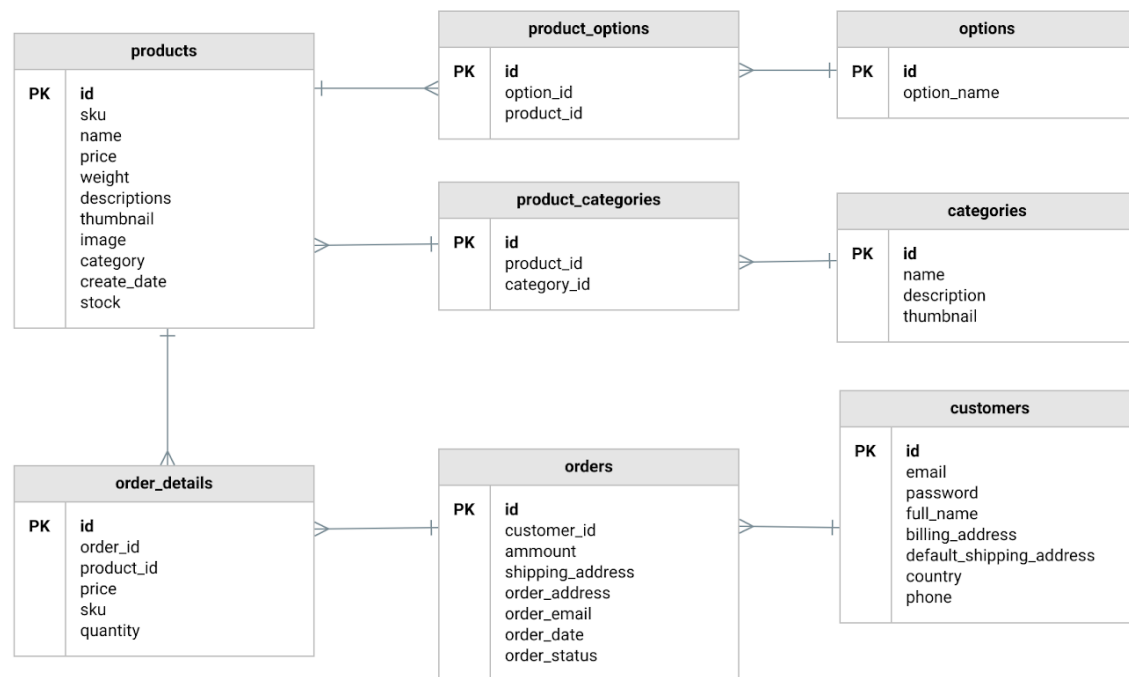
#### R.5.1: Become a seller

- Description:user can be a seller
- Input:click on signup and fill details
- Output:message("account created successfully")

#### R.5.2: Add product

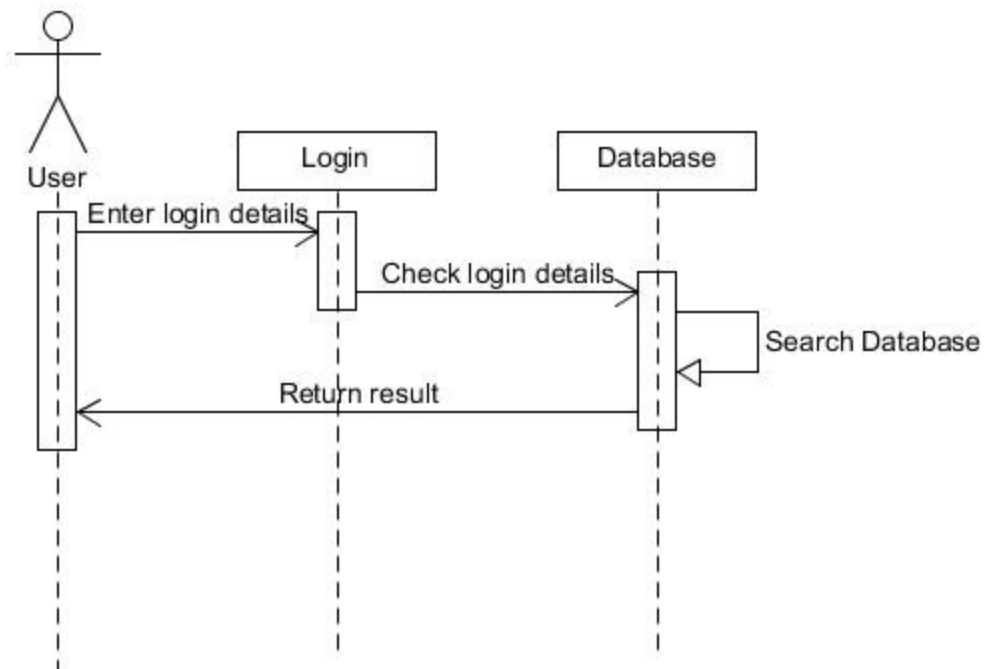
- Description:seller can add product
- Input:click on add product and fill product details
- Output:product is added and shown in product list.

## CLASS DIAGRAM

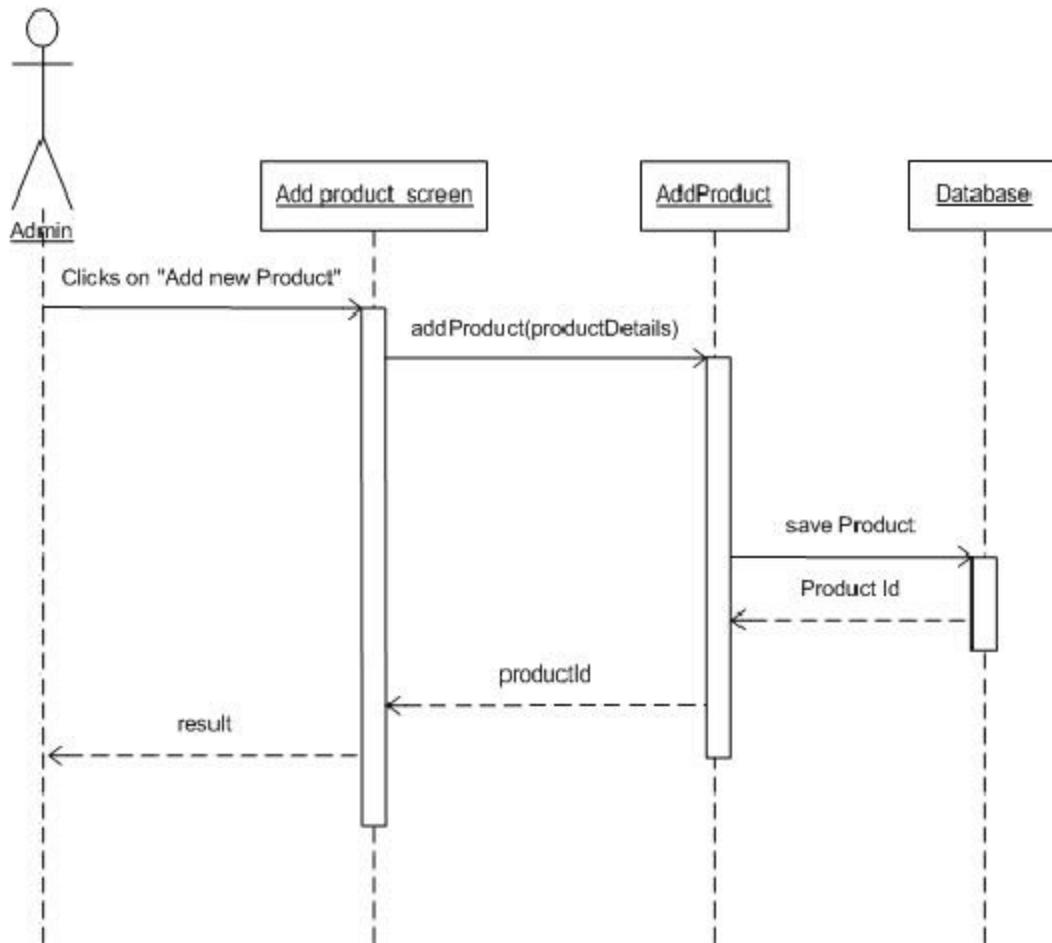




## SEQUENCE DIAGRAM



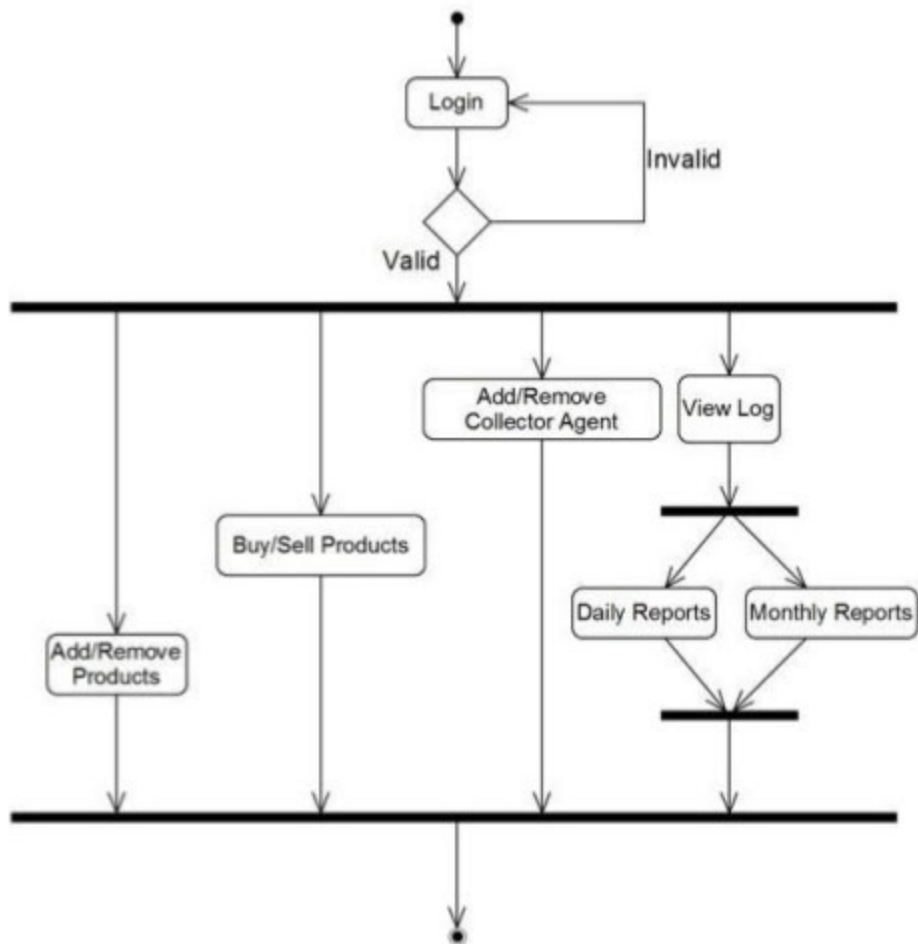
Login Sequence Diagram



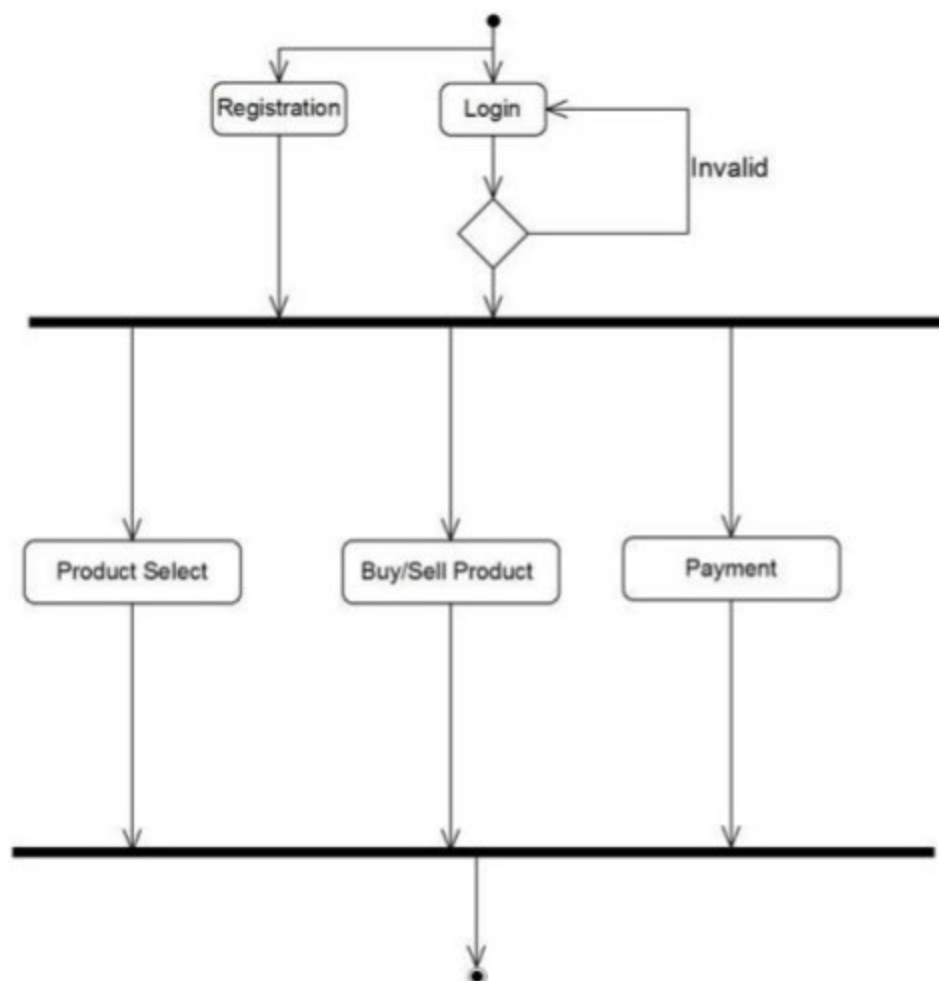
Sequence Diagram : Add product

## ACTIVITY DIAGRAM

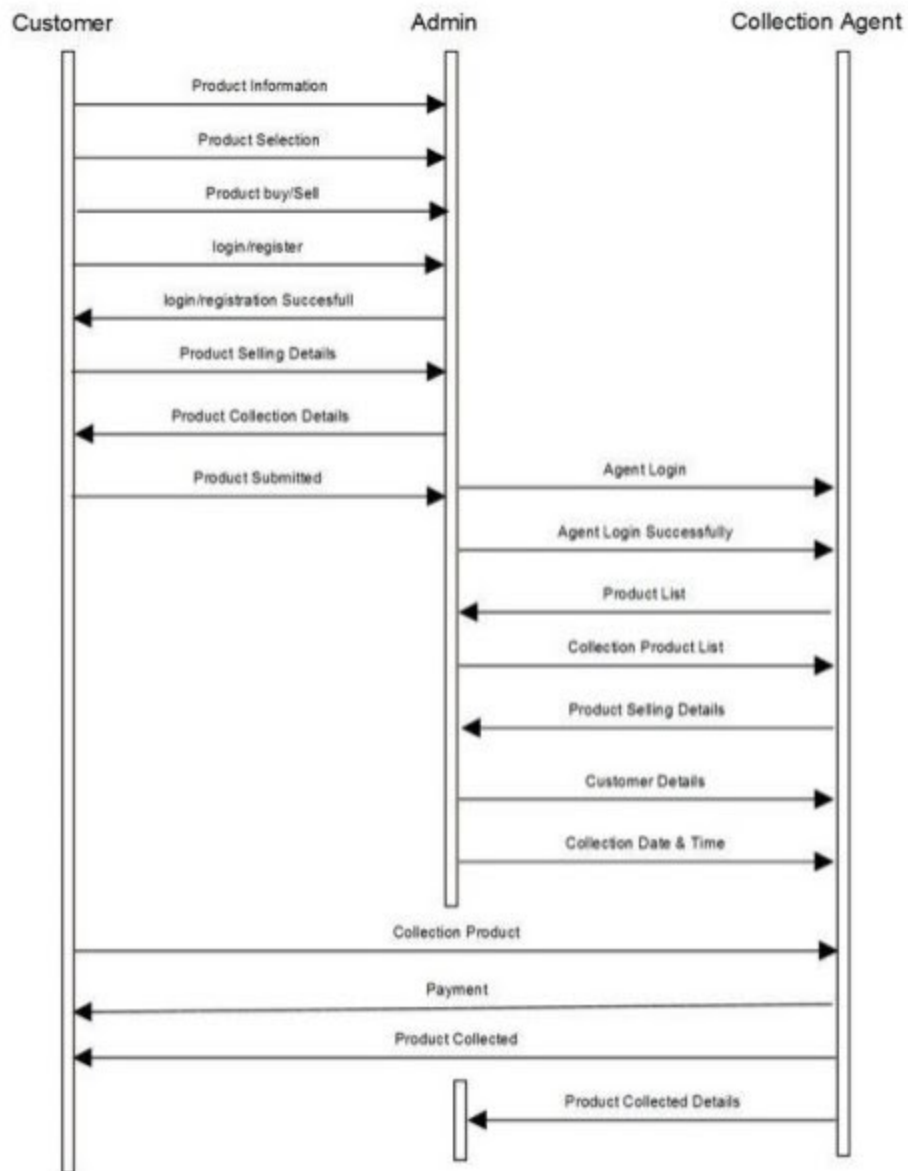
- Admin



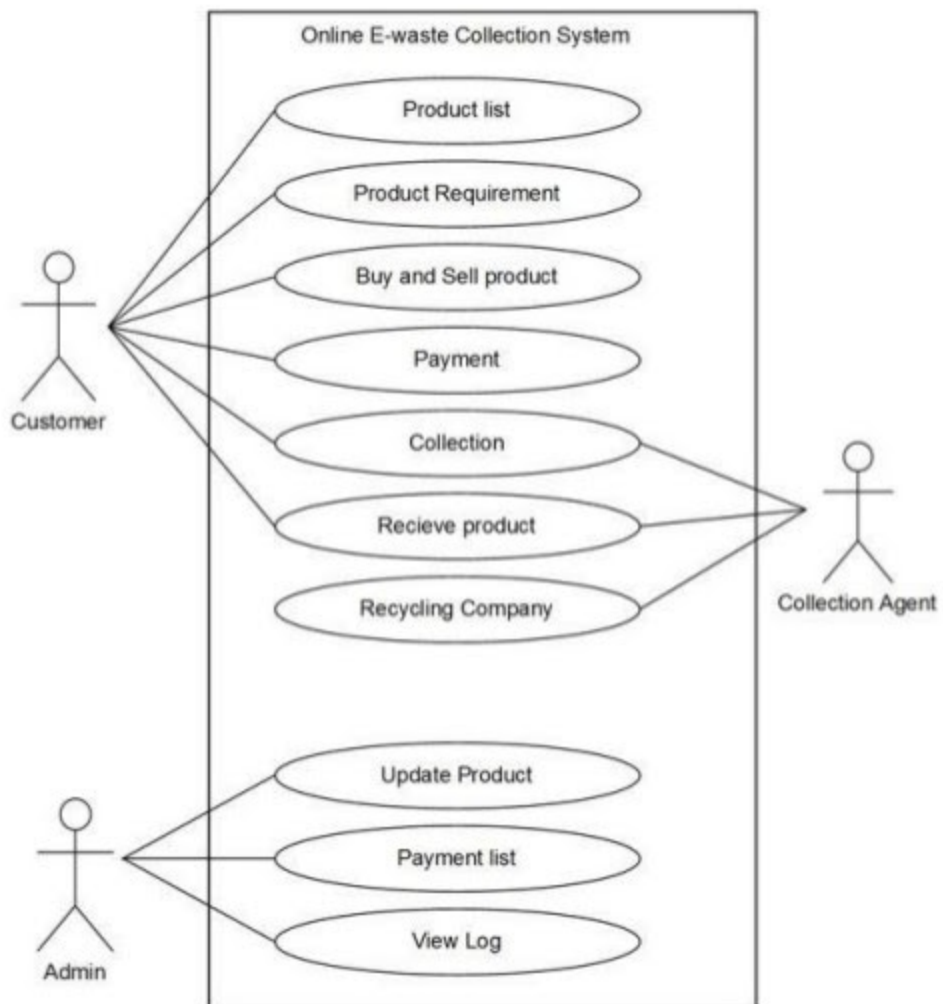
- Customer



## FLOW DIAGRAM



## USE-CASE DIAGRAM



## Implementation Details

Functionality:

- Login/Register
- Add product (for seller account)
- Buy product
- Search Product
- Cart
  1. Add to cart
  2. Remove from cart
  3. Show cart
- Wishlist
- Place order
- Filter Products
  1. By Category
  2. By Brand

### **SCREENSHOTS WITH IMPLEMENTATION:**

CART: user can add product to cart and also remove product from cart.

```

#add product to cart
@login_required(login_url='/myapp/signin')
def add_to_card(request,id):
    cart=Cart(request)
    product=products.objects.get(id=id)
    cart.add(product=product)
    if 'total' not in request.session:
        request.session['total']=0
    request.session['total']+getattr(product,'price')
    messages.success(request,"product added to cart")
    return redirect('myapp:cart')

#remove product from cart
@login_required
def item_clear(request,id):
    cart=Cart(request)
    product=products.objects.get(id=id)
    request.session['total']-=getattr(product,'price')
    cart.remove(product)
    return redirect('myapp:cart')

#clear everything from cart
@login_required
def cart_clear(request):
    cart = Cart(request)
    cart.clear()
    request.session['total']=0
    return redirect('myapp:cart')

```















```

6 | #sessions used to make a cart
7 | class Cart(object):
8 |
9 |     def __init__(self, request):#constructure of the class
10 |         self.request = request
11 |         self.session = request.session
12 |         cart = self.session.get(settings.CART_SESSION_ID)
13 |         if not cart:
14 |             # save an empty cart in the session
15 |             cart = self.session[settings.CART_SESSION_ID] = {}
16 |         self.cart = cart
17 |
18 | # methods to change cart
19 | > def add(self, product, quantity=1, action=None):...
57 |
58 | > def save(self): ...
63 |
64 | > def remove(self, product): ...
72 |
73 | > def decrement(self, product): ...
84 |
85 | > def clear(self): ...
89 |
90 |

```

## Shopping Cart

ICTS

		PRODUCT	PRICE	QUANTITY	TOTAL
		Realme 7	9999.0	 1 	9999.0
		Redmi note 10	9999.0	 1 	9999.0
		Hp monitor	5654.0	 1 	5654.0
<div> <div>Coupon:</div> <div><input type="text" value="Coupon code"/></div> <div>APPLY COUPON</div> <div>CHECKOUT</div> </div>					

**Add Product:** a seller can add product manually to our system. Those products are visible to consumer accounts.

```
@login_required
def addProduct(request):
    if request.method == 'POST':
        imageForm=ImageHandler(request.POST,request.FILES)
        if imageForm.is_valid():
            name=request.POST.get('name')
            brand=request.POST.get('brand')
            description=request.POST.get('description')

            longDescription=request.POST.get('longDescription')
            catageory=request.POST.get('catageory')
            subCatageory=request.POST.get('subCatageory')
            quantity=request.POST.get('quantity')
            image=imageForm.cleaned_data['image']
            price=request.POST.get('price')
            id=request.POST.get('product_id')

            p=products(name=name,price=price,id=id,seller=request.user,description=description,longDescription=longDescription,catageory=catageory,subCatageory=subCatageory,quantity=quantity,rating=0,brand=brand,image=image)

            p.save()
            messages.success(request,"product added successfully")
            return render(request,'myapp/index.html')

        else:
            return render(request,'myapp/addProduct.html')
```

ID   
 Product Name   
 brand   
 Price   
 Description   
 Long Description   
 category   
 sub\_category   
 Quantity   
 image  
 No file chosen

**Place Order:** user can add products to cart and place order.

```
def place_order(request):
    if request.method == "POST":
        full_name=request.POST.get('full_name')
        phone=request.POST.get('phone')
        address1=request.POST.get('address1')
        address2=request.POST.get('address2')
        city=request.POST.get('city')
        country=request.POST.get('country')
        zipcode=request.POST.get('zipcode')
        note=request.POST.get('note')
        email=request.POST.get('email')

    obj=Order.objects.create(user=request.user, full_name=full_name,
    phone=phone, address1=address1, address2=address2, city=city
```

```

, country=country, zipcode=zipcode, note=note, email=email)

x= request.session['cart']
for key,value in x.items():
    if key == '1':

product=products.objects.get(pk=value.get('product_id'))
    obj.add_product(product)

obj.save()

return redirect('myapp:index')

```

change theme in visual studio - x Document x +

127.0.0.1:8000/myapp/checkout/

HOME SHOP PAGE SINGLE PRODUCT CART CHECKOUT CATAGEORY CONTACT Search SEARCH

SEARCH PRODUCTS

Search products...

SEARCH

PRODUCTS

Sony Smart TV - 2015  
\$700.00 \$100.00

Sony Smart TV - 2015  
\$700.00 \$100.00

Sony Smart TV - 2015  
\$700.00 \$100.00

Sony Smart TV - 2015  
\$700.00 \$100.00

BILLING DETAILS

Country: United Kingdom (UK)

First Name:

Address:

Street address

Apartment, suite, unit etc. (optional)

Town / City:

Town / City

County:

State / County

Postcode:

Postcode / Zip

Email Address:

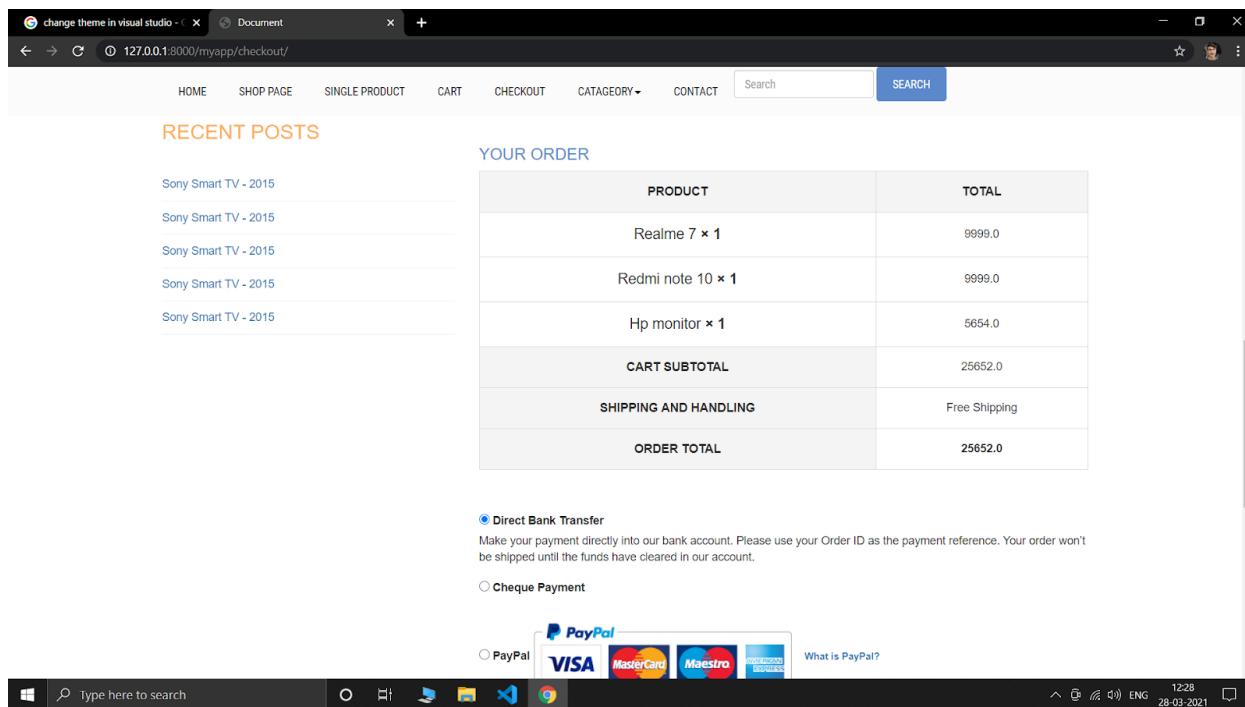
Phone:

Order Notes

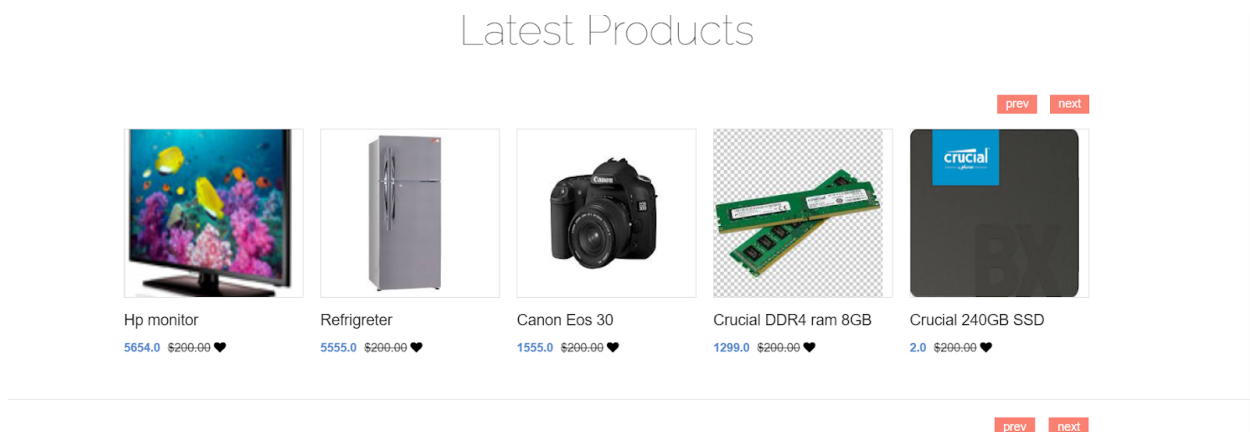
Notes about your order, e.g. special notes for delivery.

Type here to search

12:27 28-03-2021



**WishList:** it is the list where users can save favourite products(login required).then, they can see their wishlist. User have to click on heart to save to wishlist.



Show wishlist :below is a screenshot of wishlist.



Realme 7  
9999.0 \$999.00

Add to cart



Refrigerer  
5555.0 \$999.00

Add to cart



Canon Eos 30  
1555.0 \$999.00

Add to cart



Filter: user can filter products using brand name and category.

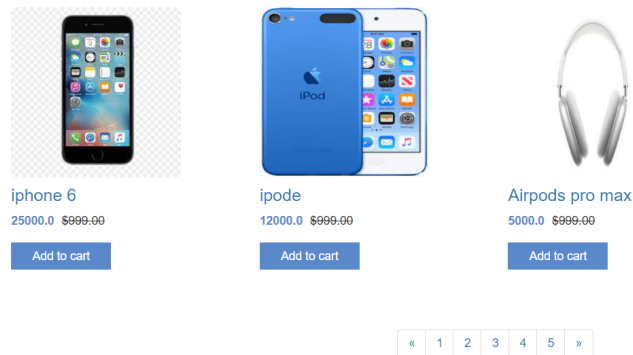
Canon

SAMSUNG

APPLE

htc

prev next



## What we Learned from this project?

### Session in django:

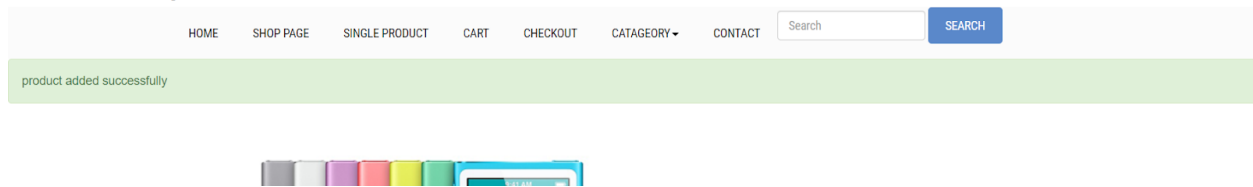
We learned session technology in django. How to create session variable and how to access them in templates. It was a difficult job to access them in views.py. We used this function in our cart.

### Django-messages:

Django has another great in-built feature called django messages. Here it

is:

```
messages.success(request,"product added successfully")
```



## Django-Login/Register:

Django provides login and register function. But ,what about when we want to change their inbuilt function. We make it in forms.py file.

```
class SignUp(UserCreationForm):
    username=forms.CharField()
    password1=forms.PasswordInput()
    password2=forms.PasswordInput()
    email=forms.EmailField()
    address=forms.CharField(max_length=200)

    class Meta:
        model=User

fields=('username','email','password1','password2','address'
)
```



```
help_texts={
    'email':None,
}

def save(self, commit=True):
    user=super(UserCreationForm, self).save(commit=False)
    user.address=self.cleaned_data['address']
    if commit:
        user.save()
    return super().save(commit=commit)

class ImageHandler(forms.Form):
    image=forms.ImageField()
```

## **Limitations And Future Extensions**

### Review system:

Giving review to the product is not implemented on the backend side.

In future it will be solved.

### Product working condition checking:

To check the quality of a product is necessary. It will be better in future.

Thanks to:

**Prof. Pinkal Chauhan**

**Prof. Jigar Pandya    Prof. Brijesh S. Bhatt**

Git : <https://github.com/dj-2002/myproject/tree/SP+SEPP>