National College of Ireland

<BSHCSD4>

<Software Development >

<2020/2021>

< Md Bellal Hossain>

< x17114918>

< x17114918@student.ncirl.ie>

<Software Project>

Technical Report

Contents

Executive Summary

Owing to the various advantages and benefits, nowadays days, ever more people assert they prefer to shop online over traditional shopping. In current history, the consumer's choice pattern has changed significantly. Without ever interacting to a sales agent, customers do thorough research online. Buyers now make more direct online and mobile transactions, never walking into typical brick-and-mortar locations.

Direct online sale expands your scope. For an online shop, the percentage of visitors who will personally visit your brick-and-mortar location no longer limits one's income. People can sell, eliminating all territorial constraints, across cities, states, even across the territories. An online store also enables people to appeal to consumers who find it easier to search and order at times where stores nationwide are not typically open.

Most individuals choose to shop online and buy items that they really do not find or are not available for purchase in their local stores. Nowadays with the aid of modern technologies as well as the support of the internet, by simply viewing in their homes, consumers from all over the world have started purchasing products online. Buying goods and products via the Web is a very simple task to do.

1.0 Introduction

1.1. Background

In current world situation people are more likely to shopping online rather than go to store physically. Some of the country's government discourage people not to go outside and to maintain their important activity online when sitting down at home. So online shopping becomes more attractive and the online market is larger than ever.

I am making a restful e-commerce web application called ‘Bismillah e-Shopping’ as my final year project. In my project we focused on to deliver a responsive, functional web application which will accede client expectations.

In our shop our target audience is anyone with small or medium size of business. The application will display various consumer product that business owner wants to sell on their shop with a responsive user interface. Any business entrepreneur can use my application for their online business with placing their own data. The application will have the backend server from where the management can maintain their product information. The user can view an item, rate the item, search the items, and buy the item with their PayPal account or visa debit/credit card. After complete purchase user will get confirmation by email and information about the item delivery. The user also can-do live chatting with the administration for any product information.

1.2. Aims

What does the project aim to achieve?

1.3. Technology

Django

Django is a moderate web-based Python platform that facilitates fast creation and tidy, pragmatic architecture. Built by professional developers, it takes care of a lot of the web development problems, so anyone can concentrate on developing the software without having to reinvent the wheel. This is a free and open-source platform

Bootstrap

Bootstrap is a CSS framework that used to design responsive and mobile first application

PayPal

PayPal used in order to enable the user to check out the chosen products. PayPal supports the global data protection and anti-fraud technologies to protect the details safe and reduce the chance of internet scams.

Jetbrains Pycharm IDE

PyCharm is an Interactive Development Platform (IDE) being used software development, especially in the Python programming language. It includes an overview of code, a graphical debugger, an automated unit tester, collaboration with version control systems (VCS) and facilitates web creation with Django.

Languages

Python

is an interpretive, object-oriented, high-level programming language with complex syntax. Its high-level built-in data structures, combined with dynamic semantics and automatic linking, make it very appealing for Accelerated software development as well as for being used as a scripting or adhesive language to link dependencies together.

JavaScript

HTML 5

CSS

Cloud platform

PythonAnywhere

PythonAnywhere is a web hosting platform to deploy the web application.

Local Host

To test the application

1.4. Structure

Overview-Introduction to the understanding of the project, what it plans to do, and why this is being created. In addition, the goals of the project and the technology and the method will be adopted and used to build the program.

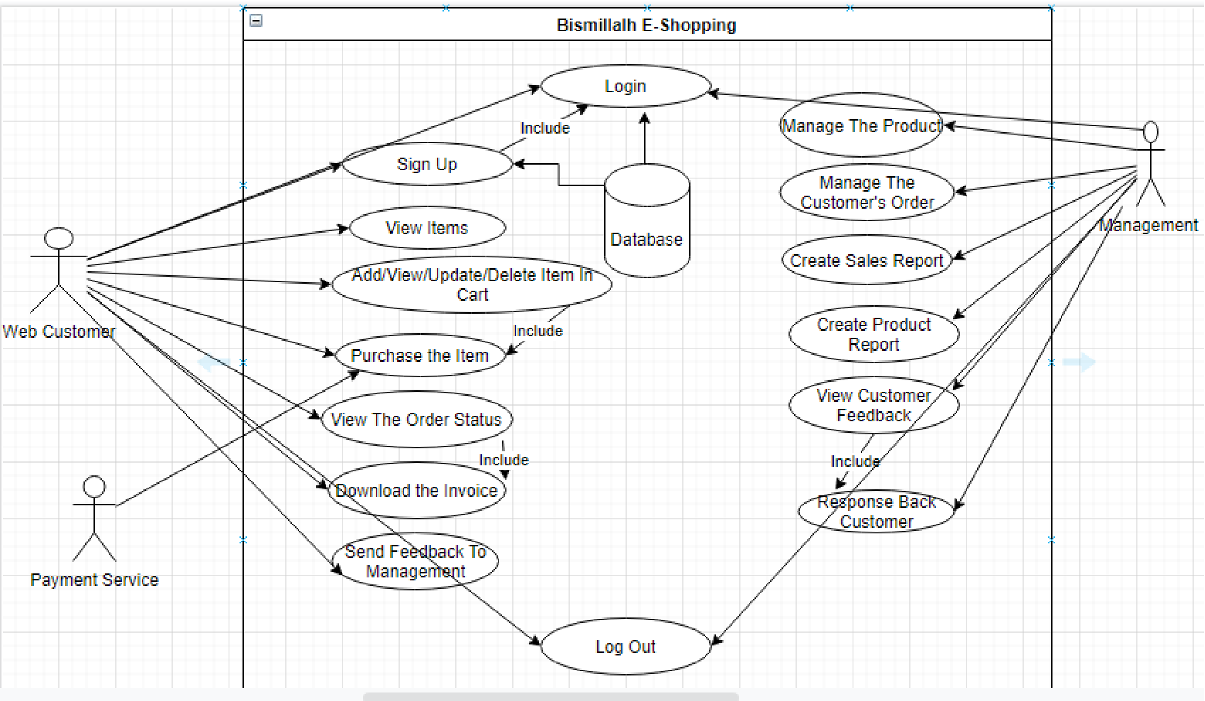
System – a summary of all structures of the application has provided in the system section of the technical report. All Functional specifications and non-functional requirements have also been established. All of them the requirements would include a short overview of the criteria and a small lampoon. How the use case is going to flow. This chapter also covers the architecture of the application with a Graphic representation and explanation. Graphic user interfaces are defined how the application would be done. The methods of research are Defined and clarified approaches.

Further research and development – In this chapter, the article explains how this is achieved if the system could change and how many other aspects could expand. What may be too reached if there was a longer span of time and resources available and dedicated to the developing the program. It also defines the future sectors of the sector the program could be tailored to.

2.0 System

2.1. Requirements

2.1.1. Functional Requirements



2.1.1.1. Use Case Diagram

2.1.1.2. Requirement 1 <Login>

2.1.1.3. Description & Priority

This requirement essential for Management to control the products. Without login admin cannot edit the product information. The user can view the product without login but to make payment user must need to login the system.

2.1.1.4. Use Case

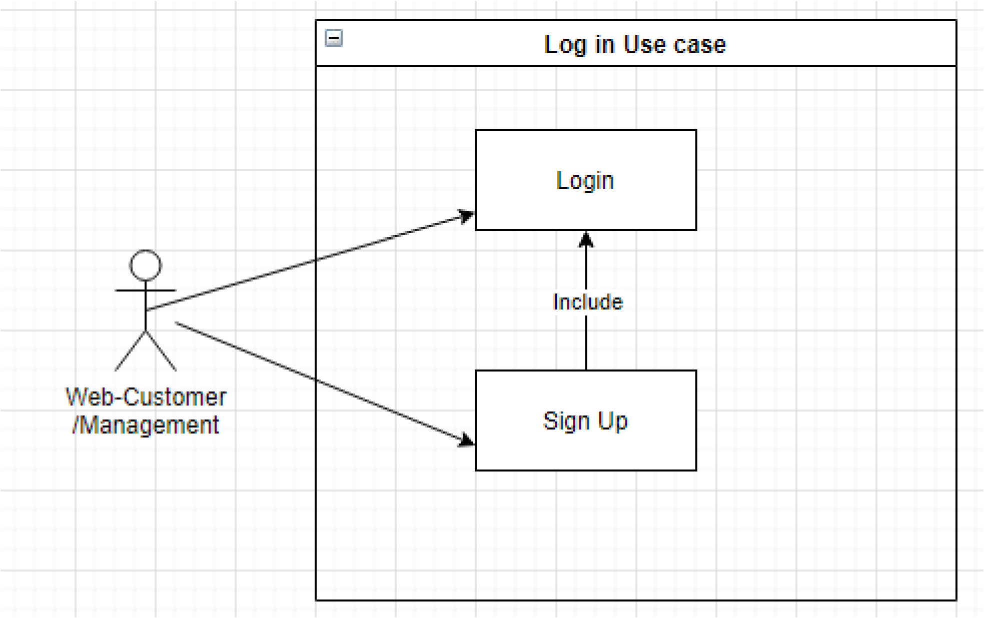
Scope

This use case is to allow the user to login the system

Description

Once the user wants to make an action, they must go through the login system. The customers are also obligated to login to make a secure payment

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode, user is ready to get on the system

Activation

This use case starts when an <User > click the signup or login button

Main flow 1

1. The system identifies that the user sends the request to sign up in the application
2. The <User> enter the credentials
3. The <User> click on signup button
4. Signup successful
5. <User> leaded to Login page

Main flow 2

1. The system identifies that the user sends the request to log in the application
6. The <User> enter the username
7. The <User>enter the password
8. The <User> click on login button
9. Login successful
10. <User> leaded to Dashboard

Alternate flow

A1 : User credentials does not maintain the criteria

1. The <user> entered the credentials is not fulfil the criteria
2. The system asks the user to try again with correct format of credentials
3. Alternate flow 1 ends

A2:<User does not have an account >

1. The System asks user to sign up first to log in the system
2. The user repeats the main flow 1

Exceptional flow

E1 : <User already has an account with same username>

1. The system verified that user credentials is already in used
2. The system asks the user to login the system or enter different credential to sign up

E1 : <User forgot the credentials>

1. The <User> forgot the username/password
2. The system asks the user to re-create the credentials

Termination

The system presents the customer/admin dashboard if user successfully logged in the system. The login process terminates here.

Post condition

Success Condition:

1. User success to login the system

Failure Condition:

1. User fails to access the system

2.1.1.5. Requirement 2 <View Items >

2.1.1.6. Description & Priority

This use case essential to customers as they will want to view an item and read the item information before they buy it.

2.1.1.7. Use Case

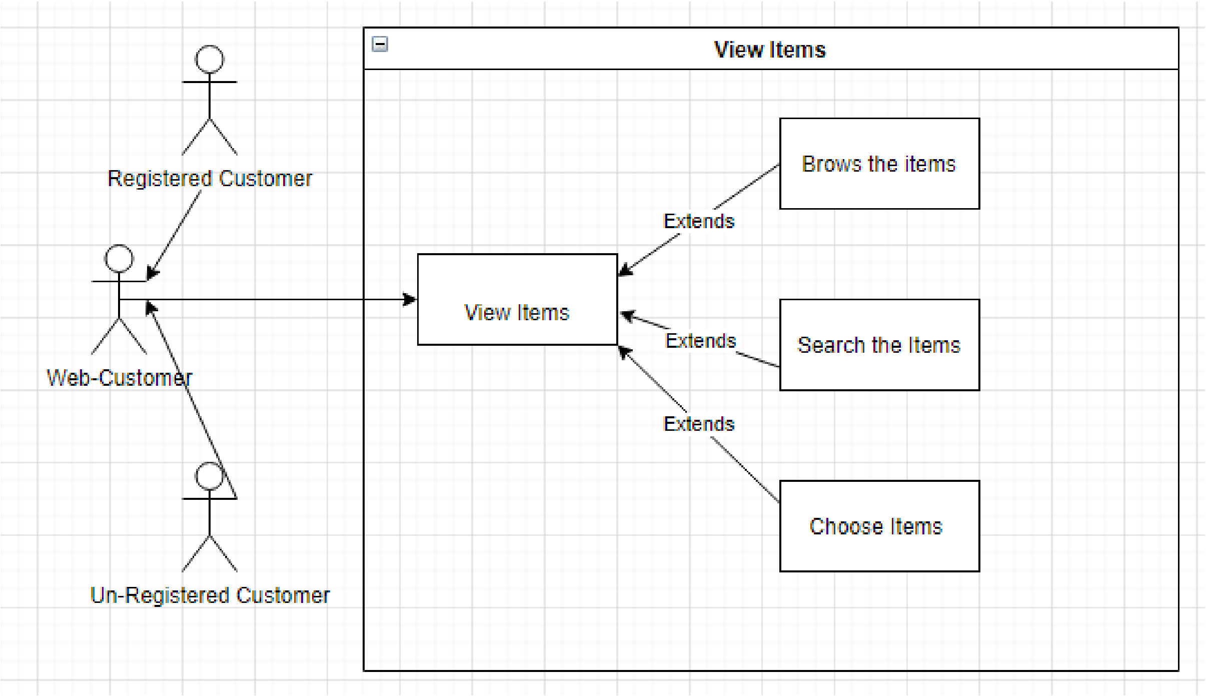
Scope

This use case is to allow customer to justify the product they want to buy.

Description

Once customer brows the system the customer dashboard will be appeared to them to choose their product to make a purchase

Use Case Diagram



Flow Description

Precondition

The system is in business mode

Activation

This use case starts when an <User> Click on a item

Main flow

1. The system displays all available products in database
2. The <User> select the item to view
3. Main flow ends

Alternate flow

A1: <User search an Specific item from the list>

1.The <User> enter a keyword to search an item in the search bar
2.The <User> click the search button
3. The <System> display all the item available in range
4. The <User> select the item to view

Exceptional flow

E1: <The system could find any product in range of users search >

1.The system will show the message “No Product found”
2.The <User>

Termination

The System displays the product stored in data database

Post condition

Success Condition:

2. User success to login the system

Failure Condition:

1.User fails to access the system

2.1.1.8. Requirement 3 <Add/View/Update/Delete Item in Cart >

2.1.1.9. Description & Priority

This use case essential to customers as they will need to choose the item and place in shopping cart in order to make a purchase.

2.1.1.10. Use Case

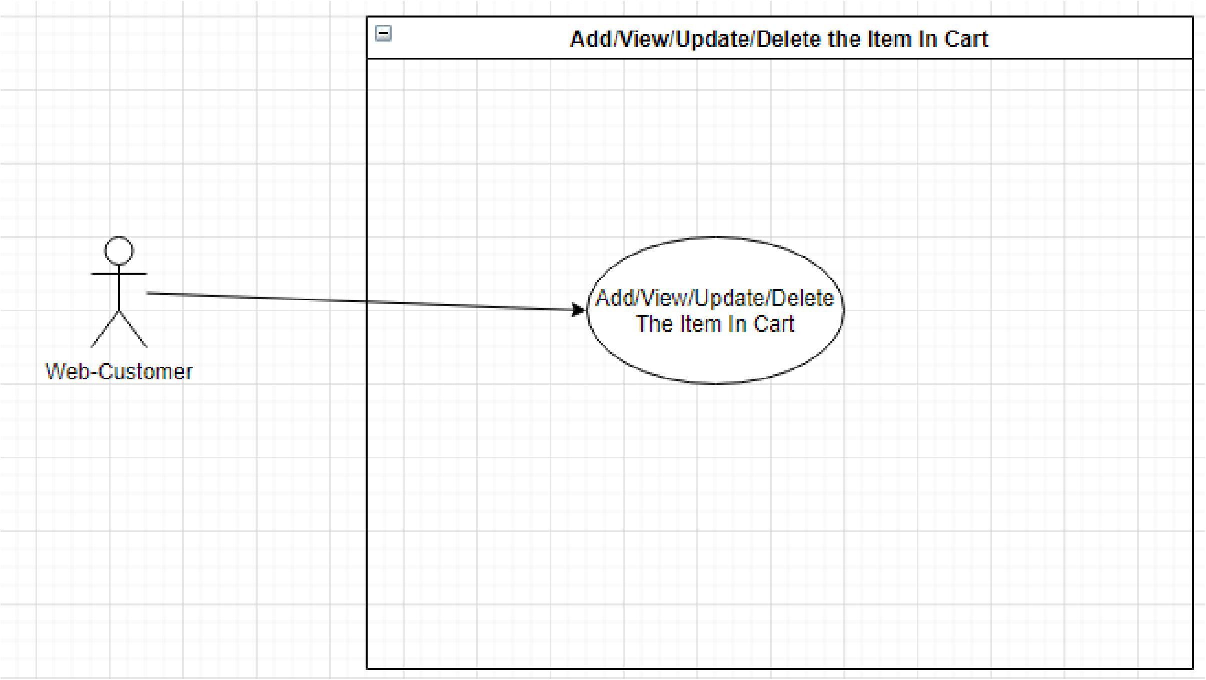
Scope

This use case is to allow customer to select the items to buy it.

Description

Once customer like an item and determined to buy it, then they click the "Add in Cart" button and product will be added to the cart with a pop-up message to ensure that the product has successfully added to the cart

Use Case Diagram


Add/View/Update/Delete the Item In Cart
Add/View/Update/Delete The Item In Cart
Web-Customer

Flow Description

Precondition

The system is in business mode

Activation

This use case starts when an <User> select item and it placed in shopping cart

Main flow The <User> select the item to add it in the cart

1. The system checks if the item available in store
2. The <User> click the product to place in cart
3. The product has placed in cart
4. The <User> review the item in cart
5. The user wants to update/delete the item in cart
6. Main flows ends

Alternate flow

A1: <User not adding the item in cart>

1. The <user> does not want to add the item in cart
2. Alternate flow ends

Exceptional flow

E1 : <The system gives warning that item not in store >

1.The system will show the message “The item is not in store”

2. Exceptional flow ends

Termination

The System add the product in cart and displays the purchase option

Post condition

Success Condition:

1.User success to add the product in cart

Failure Condition:

1. None

2.1.1.11. Requirement 4 <Purchase the Item >

2.1.1.12. Description & Priority

This use case verry important to for the system to make a successful payment

2.1.1.13. Use Case

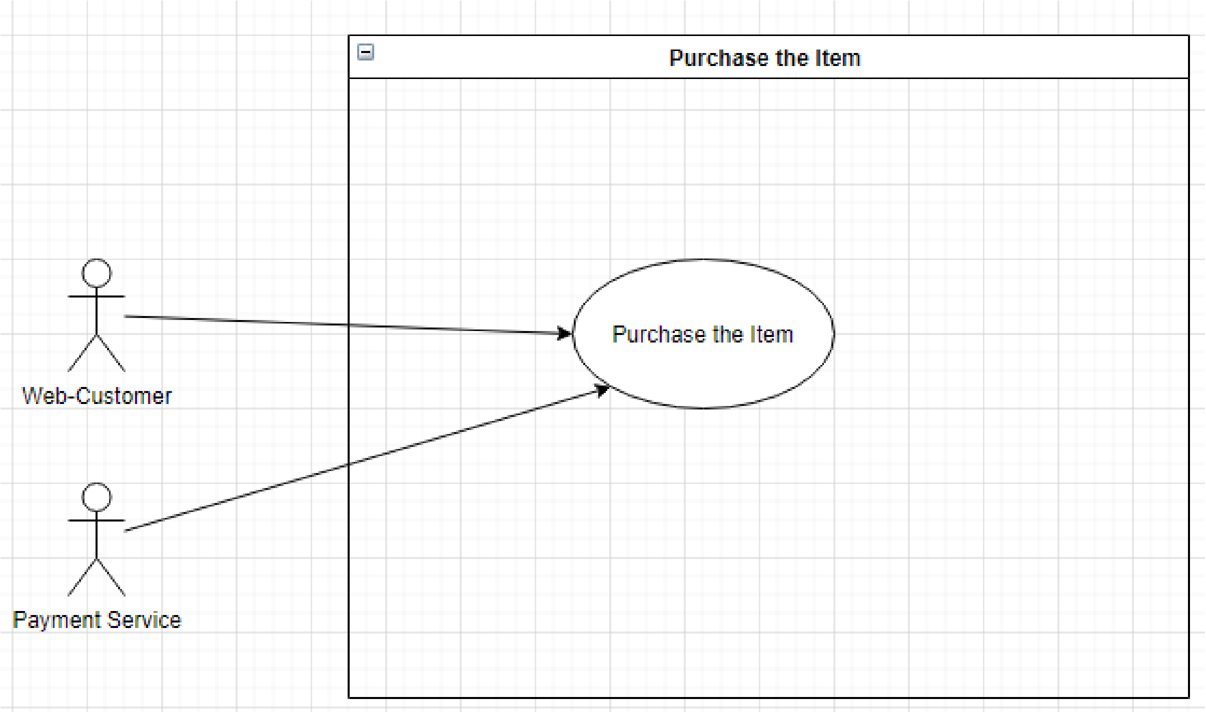
Scope

This use case is to allow customer to buy an item.

Description

Once customer click purchase button then the payment option appear to the customer to make a payment

Use Case Diagram



Flow Description

Precondition

The system is in business mode

Activation

This use case starts when an <User> want to click purchase button

Main flow

1.The <User> Click the purchase button.
2.The system bring the user to payment page
3.The <User> enter payment information
4.The <User> click Payment Button
5.Payment Successful
6.main flow ends

Alternate flow

A1 : <Payment Service refuse the payment>

1. The <System> could not verify the user payment information
2. The <System> ask the user to enter the correct information and try again
3. Alternate flow ends

Exceptional flow

E1: <The system can not connect with payment gateway >

1.The system will show the message “Cannot connect to the server”
2. Exceptional flow ends

Termination

The user made a successful payment

Post condition

Success Condition:

1.User success to buy the products trough payment

Failure Condition:

1.User could not buy the product

2.1.1.14. Requirement 5 <View the Order Status/Download the Invoice >

2.1.1.15. Description & Priority

This use case for viewing the order details and download the invoice

2.1.1.16. Use Case

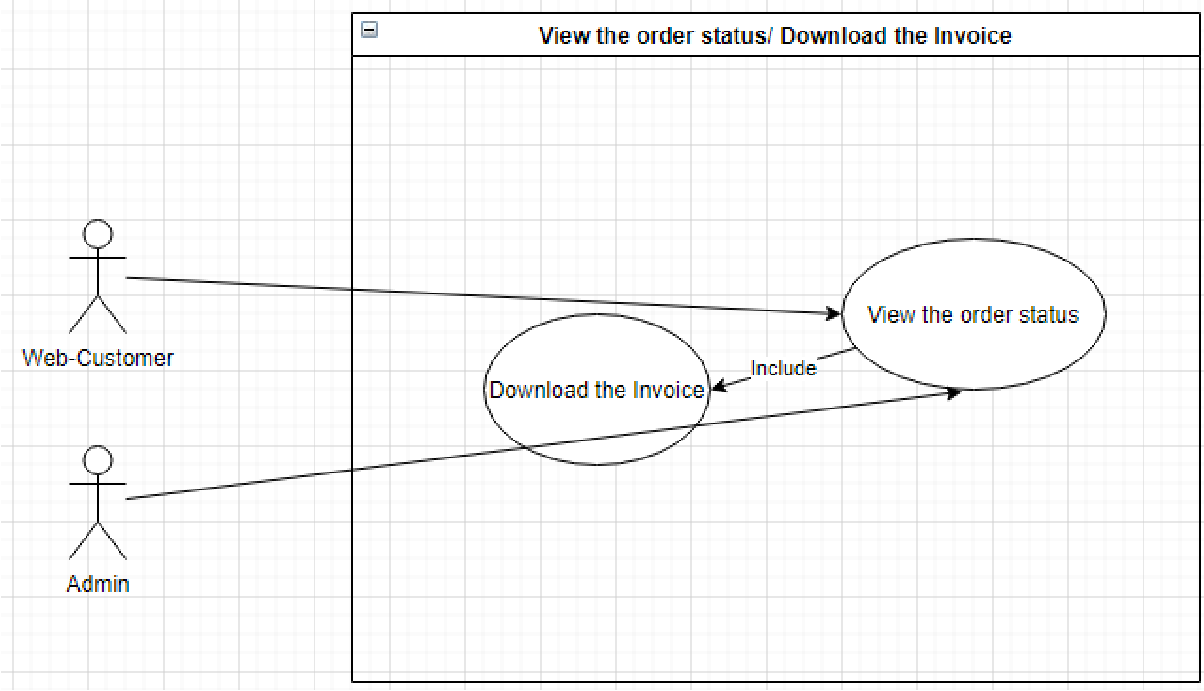
Scope

This use case is to allow customer to see the order details and download the invoice

Description

Once customer buy a product then customer can check the order status and order details by clicking the option provided and also can download the invoice for their own records

Use Case Diagram



Flow Description

Precondition

The system is in business mode

Activation

This use case starts when an <User> click the order button

Main flow

1.The <User> Click the order button.
2.The system print the order details to customer
3.The <User> view the order details
4.The <User> click the download button to download it
5.invoice downloaded
6.main flow ends

Alternate flow

A1 : <User does not have a order record>

1. The System could not find any order record for customer
2. Alternate flow ends

Exceptional flow

None

Termination

The user downloaded the Invoice

Post condition

Success Condition:

1.User success to download the invoice for records

Failure Condition:

1.User could not download the invoice

2.1.1.17. Requirement 6 <Send Feedback to management>

2.1.1.18. Description & Priority

This use case not as essential as other but send the service experience to the management

2.1.1.19. Use Case

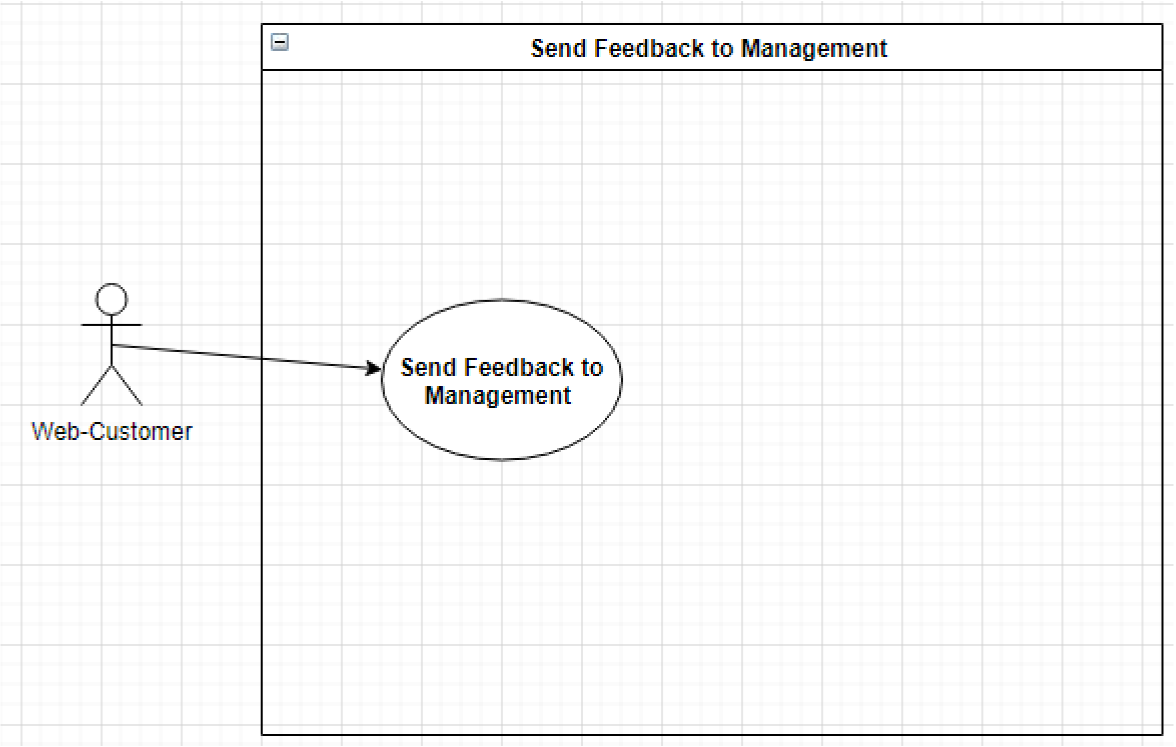
Scope

This use case is to allow customer to send the feedback to management

Description

Once customer visit the application, customer can send the feedback to management by sending direct message

Use Case Diagram



Flow Description

Precondition

The system is in service mode

Activation

This use case starts when an <User> click the send feedback button

Main flow

1.The <User> Click the “More” button.
2.The <User> choose the send feedback button
3.The <User> fill the form
4.The <User> click the send button
5.message has sent to management
6.system display the acknowledgement of sending the feedback
6.main flow ends

Alternate flow

A1 : <User does fill the form correctly >
1.The System could not send the message
2. The system ask the user to fill it correctly and try again

3. Alternate flow ends

Exceptional flow

None

Termination

The system has sent the feedback to management and display the acknowledgement

Post condition

Success Condition:

1.User success to send the feedback

Failure Condition:

None

2.1.1.20. Requirement 7 <Manage the Product>

2.1.1.21. Description & Priority

This use case important for management to control the product. Only authorised person can edit the product information.

2.1.1.22. Use Case

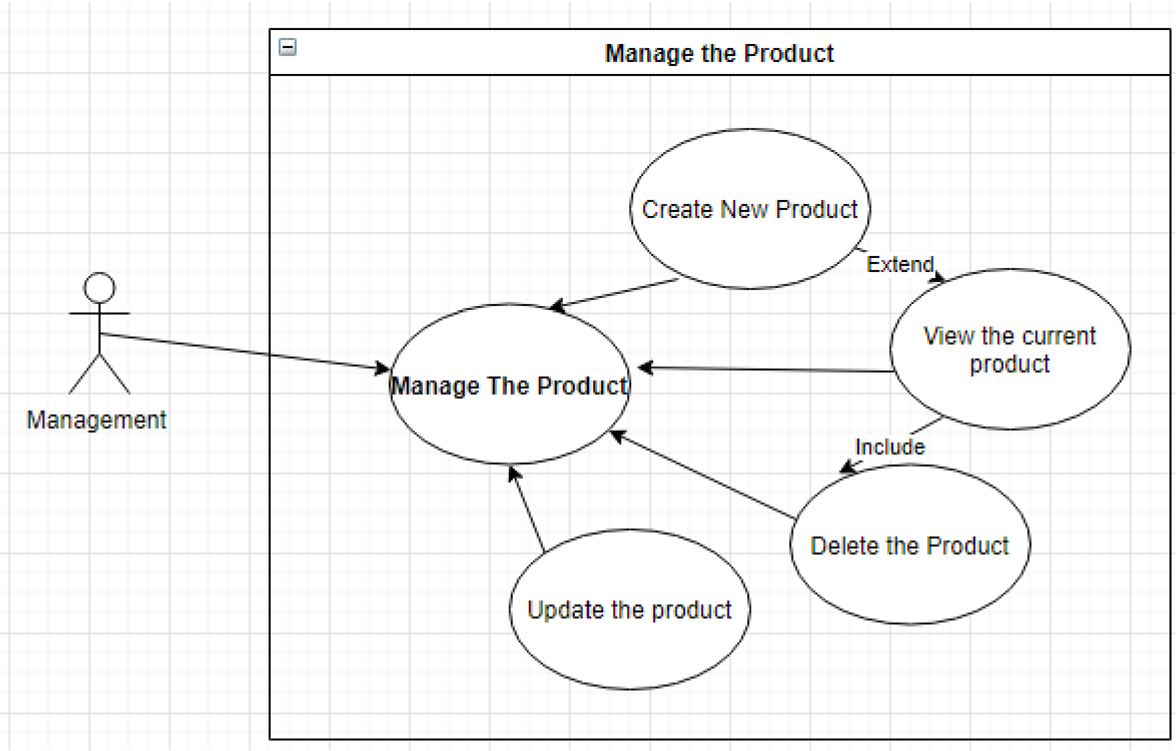
Scope

This use case is to allow admin to control the product

Description

After login into the system as an admin user can create new product, view the product, update the product and delete the product

Use Case Diagram



Flow Description

Precondition

The system is in service mode

Activation

This use case starts when an <User> click the create, update, delete button in admin dashboard.

Main flow M1

1.The <User> Click the "Add Item" button to create an item
2.The <User> enter the product information
3.The <User> click the save button
4.The <system> added the item into database
5. main (M1) flow ends

Main flow M2

1.The <User> Click the "Update" button to Update an item
2.The <User> enter update information of the item
3.The <User> click the Save button
4.The <system> update the item into database
5. main (M2) flow ends

Main flow M3

1.The <User> Click the “Delete” button to delete an item
2.The <System> ask the user if user is sure to delete the item
3.The <User> click the yes button
4.The <system> delete the item from database
5. main (M3) flow ends

Alternate flow

None

Exceptional flow

None

Termination

The system updates the database as user made an action

Post condition

Success Condition:

1.Admin success to do CRUD operation

Failure Condition:

None

2.1.1.23. Requirement 8 <Manage the Customer Order>

2.1.1.24. Description & Priority

This use case important to manage the consignment.

2.1.1.25. Use Case

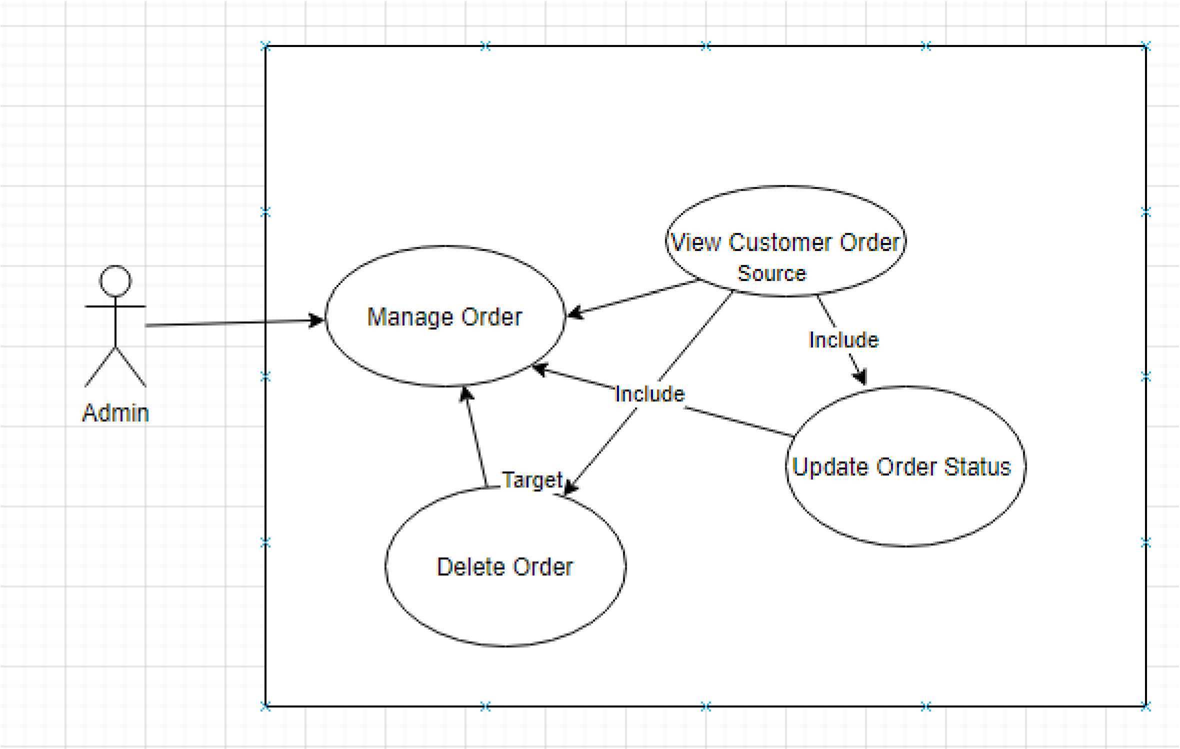
Scop

This use case is to allow admin to view the customer order and procced to order to forward.

Description

Management has authority to view the customer order. Once customer make an order, it become visible to admin portal. Admin can view the order

Use Case Diagram



Flow Description

Precondition

The system is in service mode

Activation

This use case starts when an <User> navigate to admin dashboard.

Main flow

1.The <System> display all the order that customers made
2.The <Admin> can view the orders
3. Update, Delete the order status
4. main (M3) flow ends

Alternate flow

A1<No order history>

1. There is no order to show

Exceptional flow

None

Termination

The system displays all the records of order

Post condition

Admin is updated about the recent transaction

2.1.1.23. Requirement 9 <Customer Subscription>

2.1.1.24. Description & Priority

This use case allow the user to subscribe the website for free product and promotion news .

2.1.1.25. Use Case

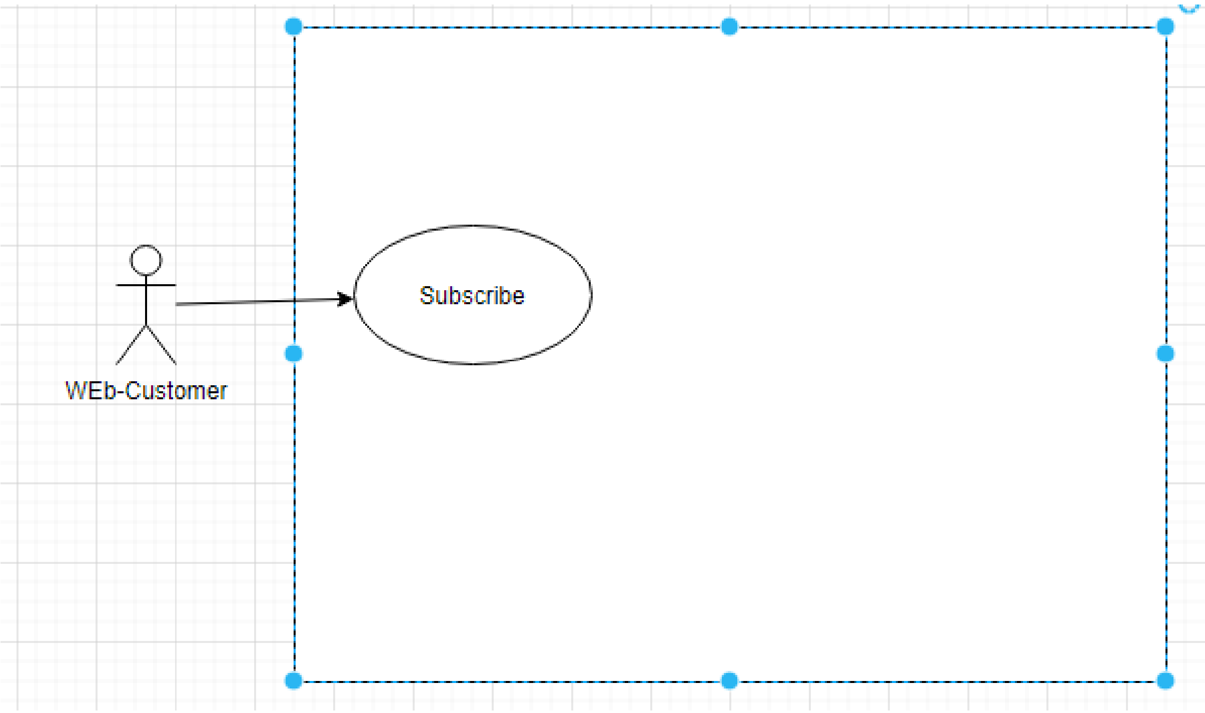
Scop

This use case is to allow admin to view the customer order and procced to order to forward.

Description

Customer can subscribe simply filling the form and sent it to admin

Use Case Diagram



Flow Description

Precondition

The system is in business mode

Activation

This use case starts when an <User> navigate to subscription tab.

Main flow

1.The <User> fill the form
2.The <User > hit the subscription button

Alternate flow

A1<No order history>

1. User hasn't enter the correct email address

Exceptional flow

None

Termination

The system will confirm the user subscriptions with email notification.

Post condition

The user confirmed about the subscription.

2.1.1.26. Requirement 10<Create Sales Report>

2.1.1.27. Description & Priority

This use case is describing the creation of the sales report

2.1.1.28. Use Case

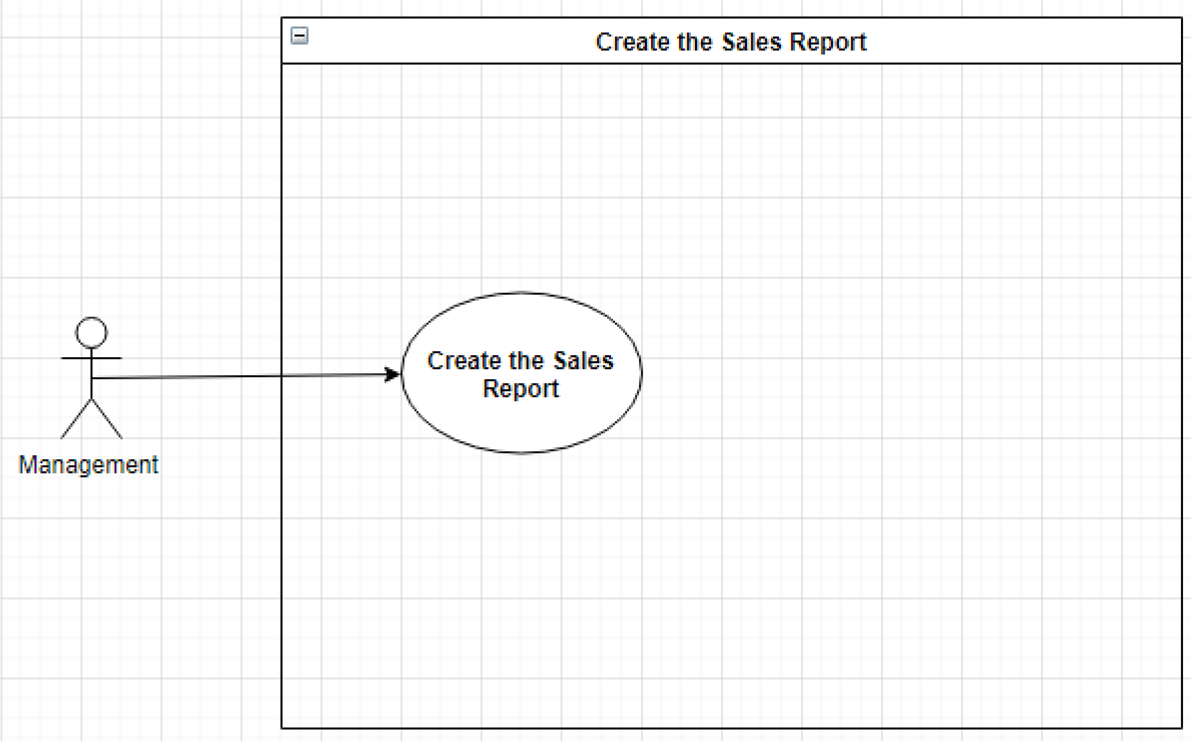
Scop

This use case is to allow admin to create the sales report

Description

Management has option to create the sales report daily, weekly or monthly

Use Case Diagram



Flow Description

Precondition

The system is in service mode

Activation

This use case starts when an <User> click the “Create Sales Report” button.

Main flow

1.The <Admin> can click on create sales report
2.The system create the report and displays to admin
2. main flow ends

Alternate flow

None

Exceptional flow

None

Termination

The system displays the sales report

Post condition

Admins gets the sales report

2.1.1.29. Requirement 11 <Create Product Report>

2.1.1.30. Description & Priority

This use case is describing the creation of the Product report

2.1.1.30 Use Case

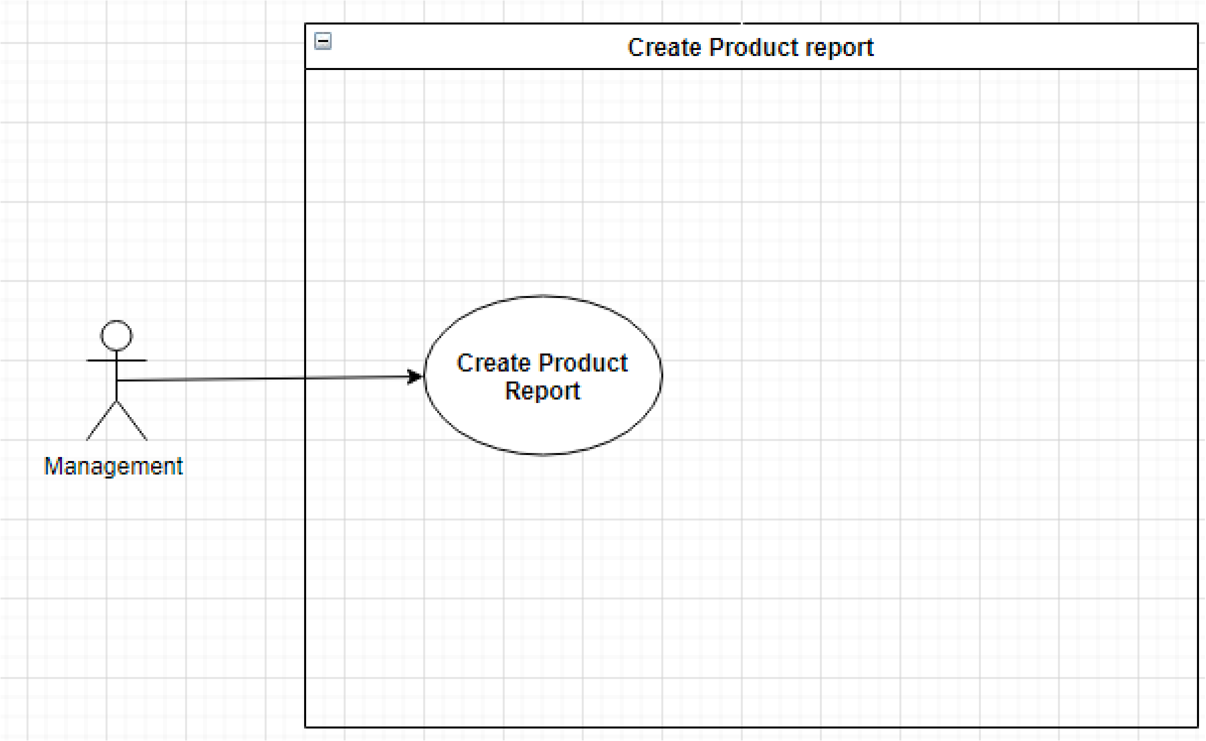
Scop

This use case is to allow admin to create the Product report

Description

Management can create product report in order to get the current stock valuation.

Use Case Diagram



Flow Description

Precondition

The system is in service mode

Activation

This use case starts when an <User> click the “Create Product Report” button.

Main flow

1.The <Admin> can click on create Product report
2.The system create the report and displays to admin
3. main flow ends

Alternate flow

None

Exceptional flow

None

Termination

The system displays the Product report

Post condition

Admins gets the Product report

2.1.1.32. Requirement 12 <Create Product Report>

2.1.1.33. Description & Priority

This use case is essential to get the customer feed back and send response back to customer

2.1.1.34. Use Case

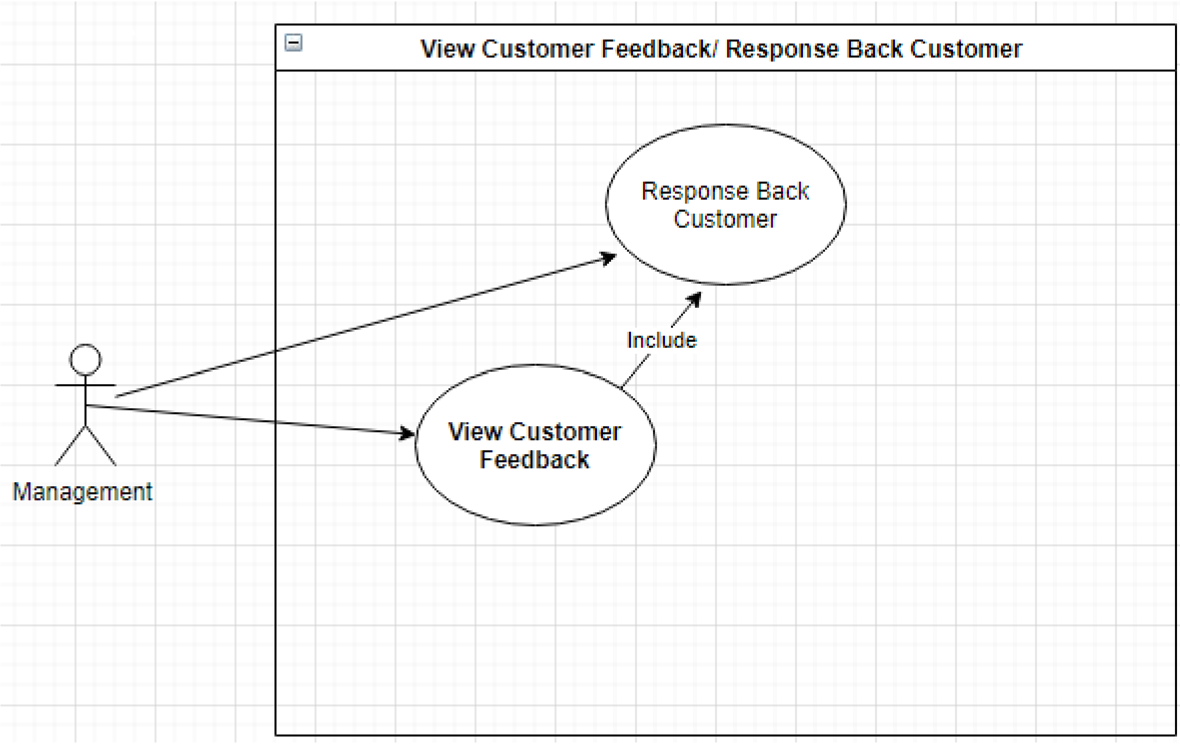
Scop

This use case is to allow admin to view the customer feedback and send the response to customer

Description

Once customer send a feedback to management, it will be visible into admin dashboard. Admin can view that and response back to customer

Use Case Diagram



Flow Description

Precondition

The system is in service mode

Activation

This use case starts when an <User> click the Feedback button

Main flow

1.The <Admin> can click on click the Feedback button to see the
2.The system displays all message to admin
3.The <User> can send response to customer
4. main flow ends

Alternate flow

None

Exceptional flow

None

Termination

The system displays all message context with sender information

Post condition

Admins need to response back

2.1.2. Data Requirements

I will use sqlit3 database system to store my store data. The database has 5 datasets in my database. The first dataset is to store all the user's information such as username, address, email, and many others. Second dataset called Products, which is for storing the products data such as product title, category, price etc. third dataset is Order to store the order information and the fourth one is to store the message from customer called Feedback. The last dataset name is customers. That will store the customer information.

2.1.3. User Requirements

End User Requirements

Customers should be able to access the application from any web browser that supports HTML 3.2 (or later) and cookies.

New entrants to the site must be able to log on their own. Users can be distinguished by special account IDs.

Transactions need to be safe. In other words, a simple authentication function should be incorporated into the program to prohibit unauthorized individuals from making purchases on behalf of the customer.

Users to the web must be able to scan the database using the appropriate keywords to find items of interest.

Users must be able to pick items of interest and add them to their shopping baskets for potential transactions.

Before checking out, customers should be able to adjust the quantity of items in and/or remove items from their shopping baskets.

All chosen products should be delivered to the customer after purchase. Customers should be able to view the status of the products they purchased.

A sufficient number of consumers should be able to access the program concurrently.

The efficiency of the application should not be degraded by an improvement in the amount of products or services provided.

Administration Requirements:

Administrators must be enabled to use web browsers to handle the application. administrators should be able to exclude users. administrators should also be able to adjust the status of products bought by customers just after product is delivered. Administrators should be able to access the payments of the customer. Site administrators should also be able to view all purchases on a day basis.

2.1.4. Environmental Requirements

The device must need to connect to internet to access the application. The application can be access from any smart device such as Smart phone, Table, computer etc.

2.1.5. Usability Requirements

Navigation and structure:

The system must have a structure that would allow users to discover the product they need with a minimum number of clicks. The simplicity of navigation allows users to spend extra time interacting, consuming its pages. In addition, a very well layout of the web store is also important for search engine optimization (SEO).

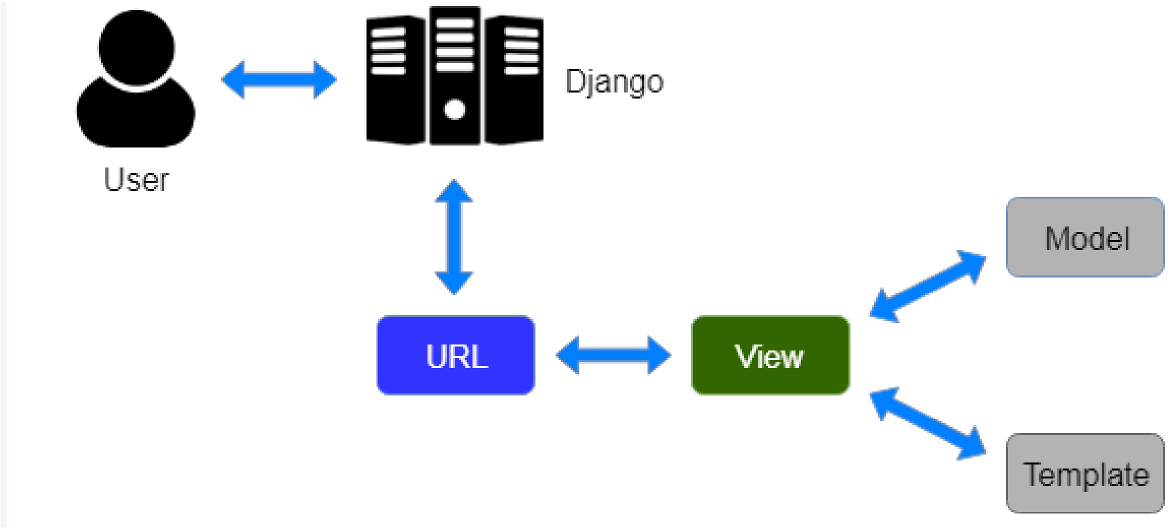
Convenience of a product catalogue:

More and more items we have in our catalogue, the more important it is. Deliver consumers with the potential of sorting items according to different criteria, like cost, date of delivery, reputation, etc. Develop and introduce product filters that allow prospective purchasers to use various choices to locate a product and give them the ability to compare various types of products.

Description and Image of the Item:

Therefore, more details the purchaser can get from the website, the better. That is indeed why viewing the picture and explanation of the item as just an essential feature of the usability of the application. All the required components must be placed on one page: comprehensive product overview, high-quality photographs, technological specs, cost, stock supply, delivery charges, payment options, etc. At any rate, the accuracy of the pictures must be reasonable, although the need to render a certain amount of photographs from various angles ought to be good.

2.2. Design & Architecture



Our application will be developed using Django framework which based on MVT (Model, View, Template) design patten. The Model facilitates the database. This is a data access layer that manages the data.

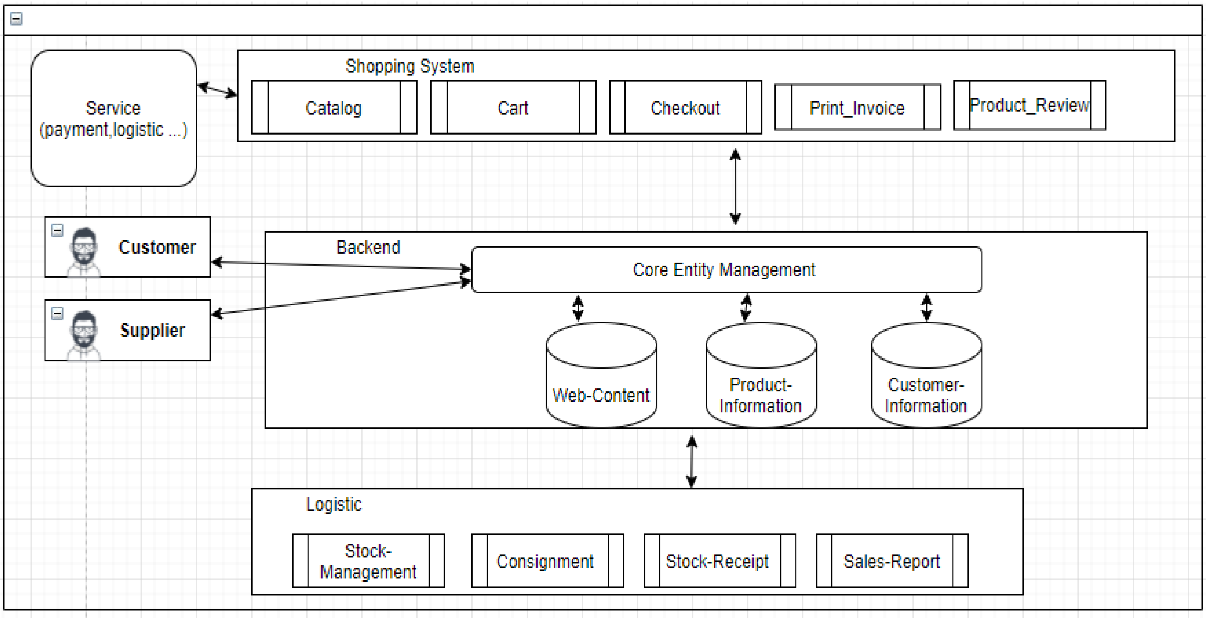
The template is a presentation layer that manages the client-side component fully. The View is used to implement business logic and communicate with a data transfer and template model.

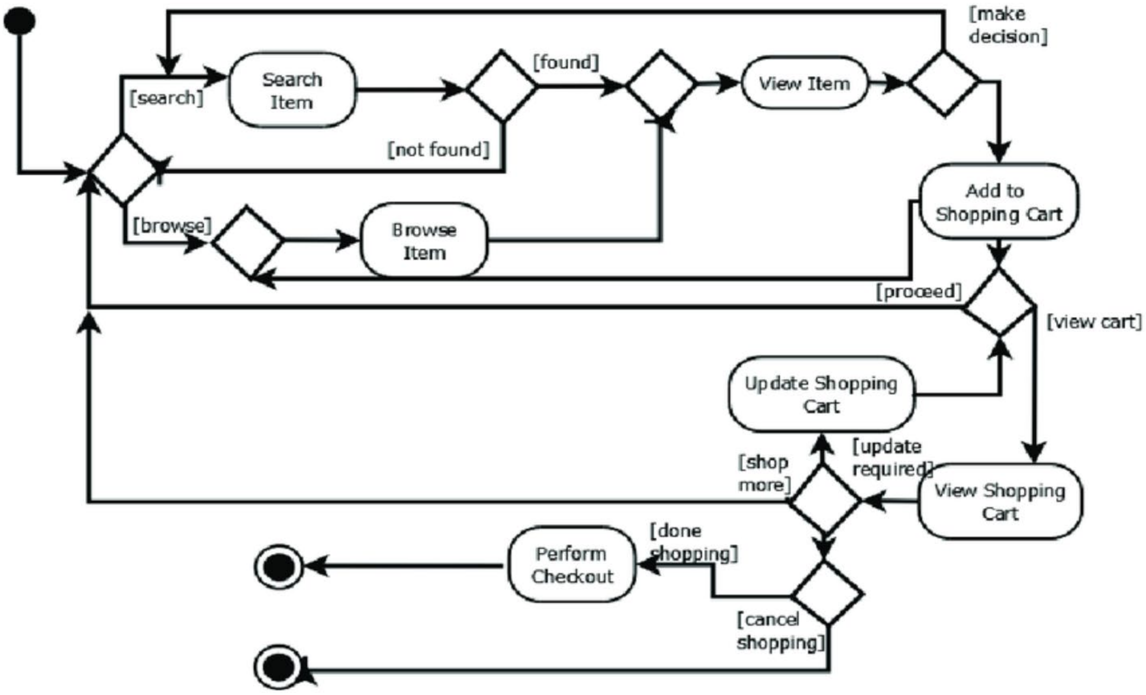
when the user requests a resource to Django, Django acts as a handler and tests it to the available resource in the URL.

If the URL maps are considered a view that communicates with the model and template, the template would be rendered.

Django replies to the user and sends a sample as a result.

The backend side of the application will be developed using pure python and the client side will be developed using HTML, CSS, Bootstrap, JavaScript.


Service (payment,logistic ...)
Shopping System
Catalog
Cart
Checkout
Print_Invoice
Product_Review
Customer
Supplier
Backend
Core Entity Management
Web-Content
Product-Information
Customer-Information
Logistic
Stock-Management
Consignment
Stock-Receipt
Sales-Report

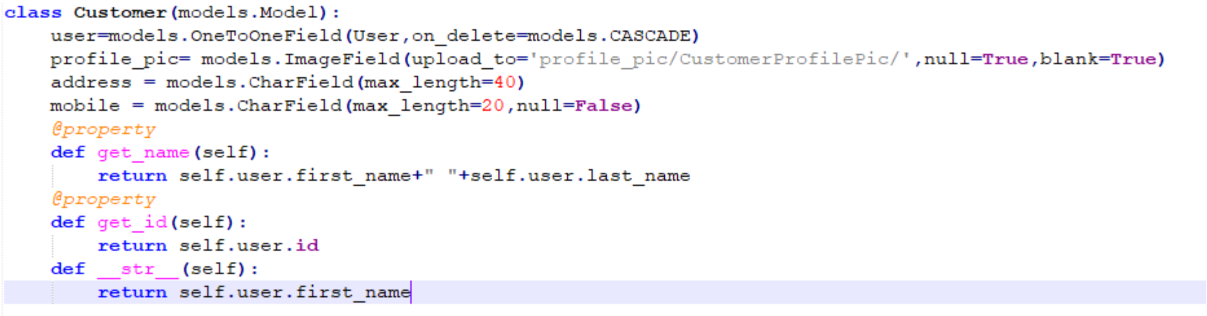

[make decision]
Search Item
[found]
View Item
[search]
[not found]
Add to Shopping Cart
[browse]
Browse Item
[proceed]
[view cart]
Update Shopping Cart
[shop more]
[update required]
View Shopping Cart
[done shopping]
Perform Checkout
[cancel shopping]

2.3. Implementation

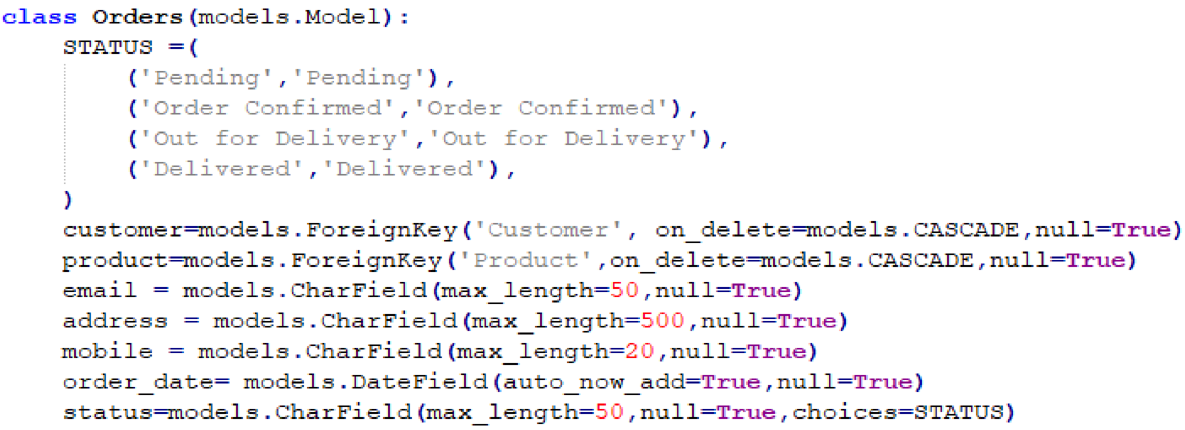
Classes

Customer class:

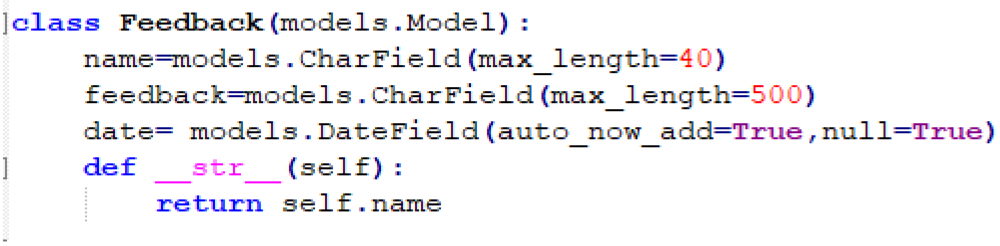
In this class I initialize the customer information in database and this information only be visible to the admin. No other user can access this information. In same way we have other classes that developed to initialize the product information, order information, customer feedback information etc.



```
class Customer(models.Model):
    user=models.OneToOneField(User,on_delete=models.CASCADE)
    profile_pic= models.ImageField(upload_to='profile_pic/CustomerProfilePic/',null=True,blank=True)
    address = models.CharField(max_length=40)
    mobile = models.CharField(max_length=20,null=False)
    @property
    def get_name(self):
        return self.user.first_name+" "+self.user.last_name
    @property
    def get_id(self):
        return self.user.id
    def __str__(self):
        return self.user.first_name
```



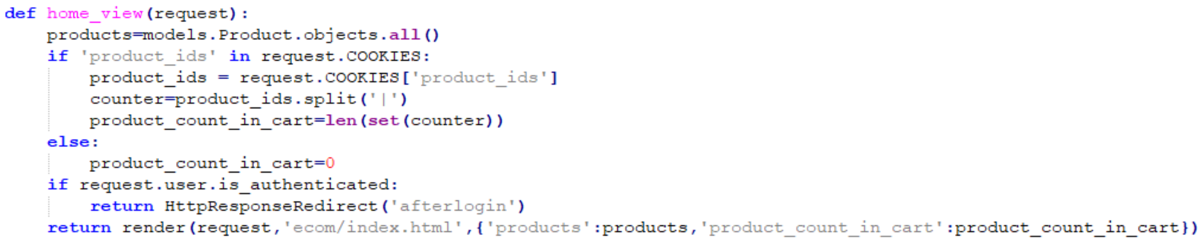
```
class Orders(models.Model):
    STATUS =(
        ('Pending','Pending'),
        ('Order Confirmed','Order Confirmed'),
        ('Out for Delivery','Out for Delivery'),
        ('Delivered','Delivered'),
    )
    customer=models.ForeignKey('Customer', on_delete=models.CASCADE,null=True)
    product=models.ForeignKey('Product',on_delete=models.CASCADE,null=True)
    email = models.CharField(max_length=50,null=True)
    address = models.CharField(max_length=500,null=True)
    mobile = models.CharField(max_length=20,null=True)
    order_date= models.DateField(auto_now_add=True,null=True)
    status=models.CharField(max_length=50,null=True,choices=STATUS)
```



```
class Feedback(models.Model):
    name=models.CharField(max_length=40)
    feedback=models.CharField(max_length=500)
    date= models.DateField(auto_now_add=True,null=True)
    def __str__(self):
        return self.name
```

Home View function:

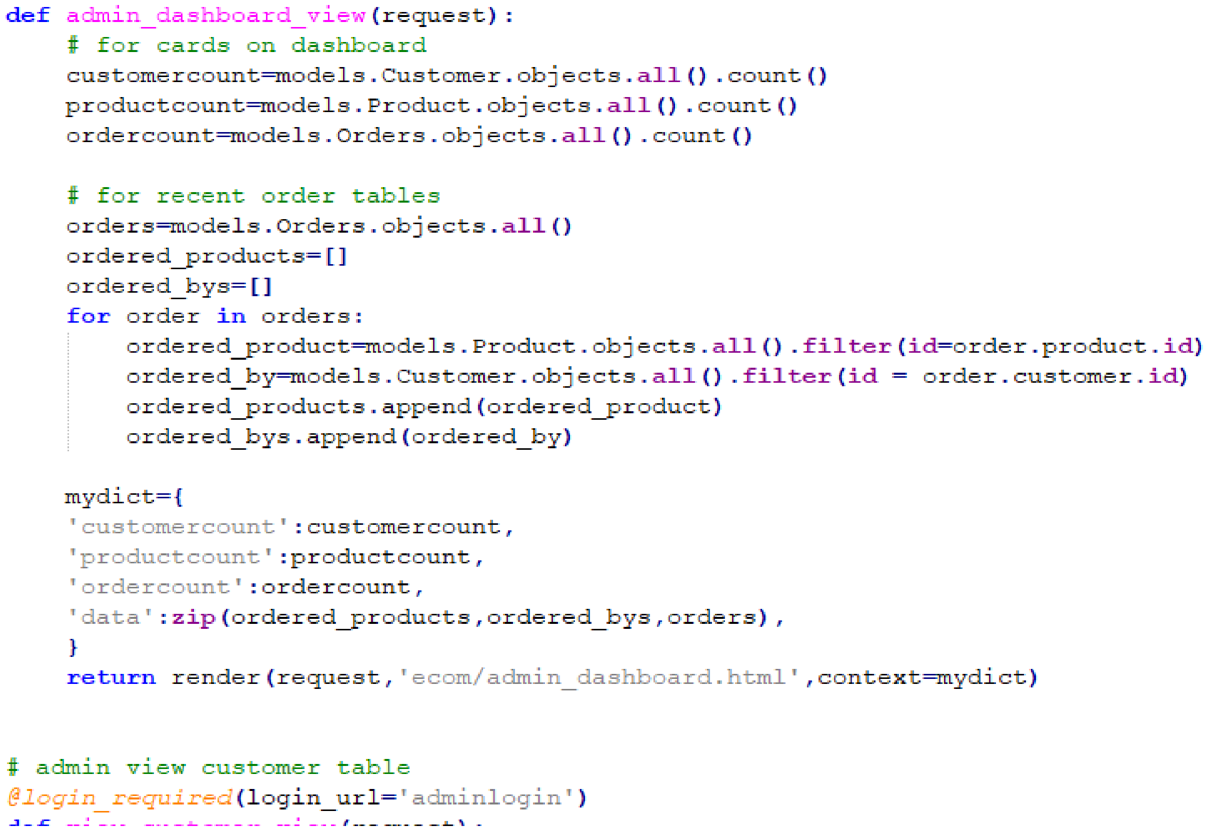
This function take a web request and send an HTML response by rendering a HTML page. In this page I try to display the product information to the end user.



```
def home_view(request):
    products=models.Product.objects.all()
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        counter=product_ids.split('|')
        product_count_in_cart=len(set(counter))
    else:
        product_count_in_cart=0
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'ecom/index.html',{'products':products,'product_count_in_cart':product_count_in_cart})
```

Admin Dashboard View function:

In this view function I developed the algorithm that get HTTP request and response a HTTP response by rendering an HTML page. In this page admin will be able to view all the products available in the stock, all the orders that customers made and all the message that customers sent. The function call a HTML page call admin dashboard.html that decorate the interface and retrieve the data from database by help of associated view function.

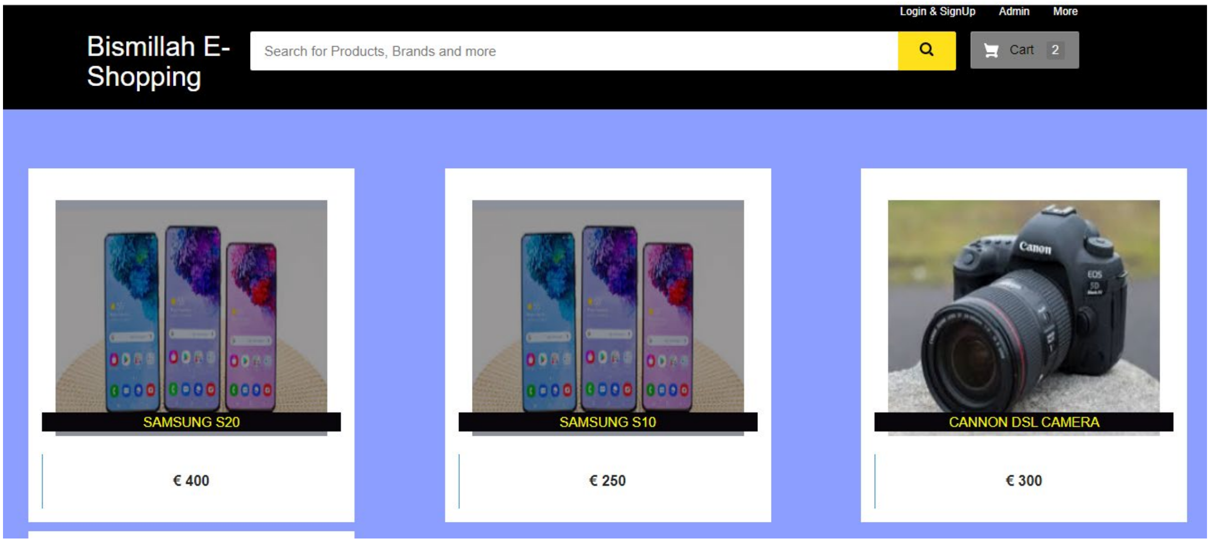


```
def admin_dashboard_view(request):
    # for cards on dashboard
    customercount=models.Customer.objects.all().count()
    productcount=models.Product.objects.all().count()
    ordercount=models.Orders.objects.all().count()

    # for recent order tables
    orders=models.Orders.objects.all()
    ordered_products=[]
    ordered_bys=[]
    for order in orders:
        ordered_product=models.Product.objects.all().filter(id=order.product.id)
        ordered_by=models.Customer.objects.all().filter(id = order.customer.id)
        ordered_products.append(ordered_product)
        ordered_bys.append(ordered_by)

    mydict={
    'customercount':customercount,
    'productcount':productcount,
    'ordercount':ordercount,
    'data':zip(ordered_products,ordered_bys,orders),
    }
    return render(request,'ecom/admin_dashboard.html',context=mydict)


# admin view customer table
@login_required(login_url='adminlogin')
def view_customer_view(request):
```

2.4. Graphical User Interface (GUI)

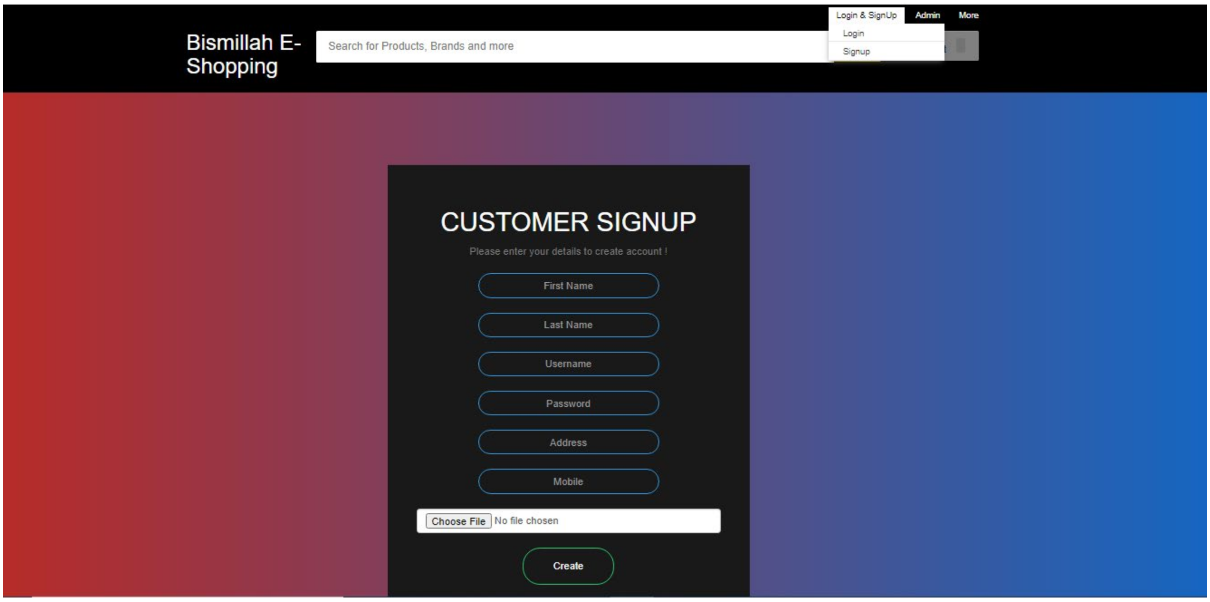
2.4.1 Home View

In the home view interface, the application displays all available products to customer with the product information and price. Users have the option to place the product in cart. The add to cart option will appear to customer when user hoover the mouse over the product



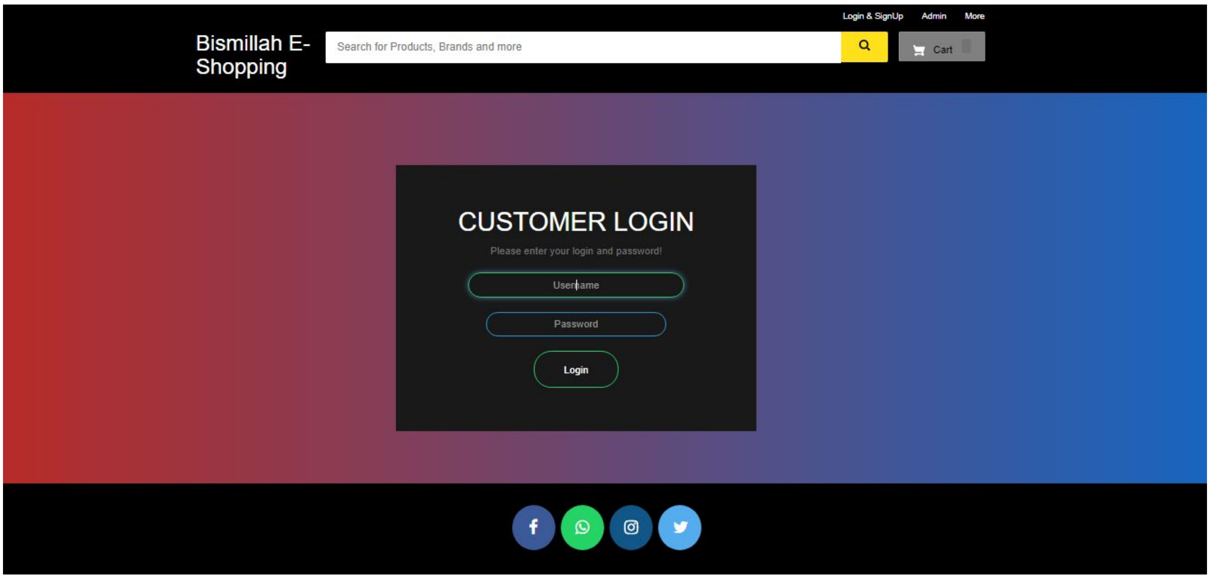
2.4.2 Customer Signup Page

This page facilitate the customer or admin to sign up the application with their information such as name, email, address, phone number etc.



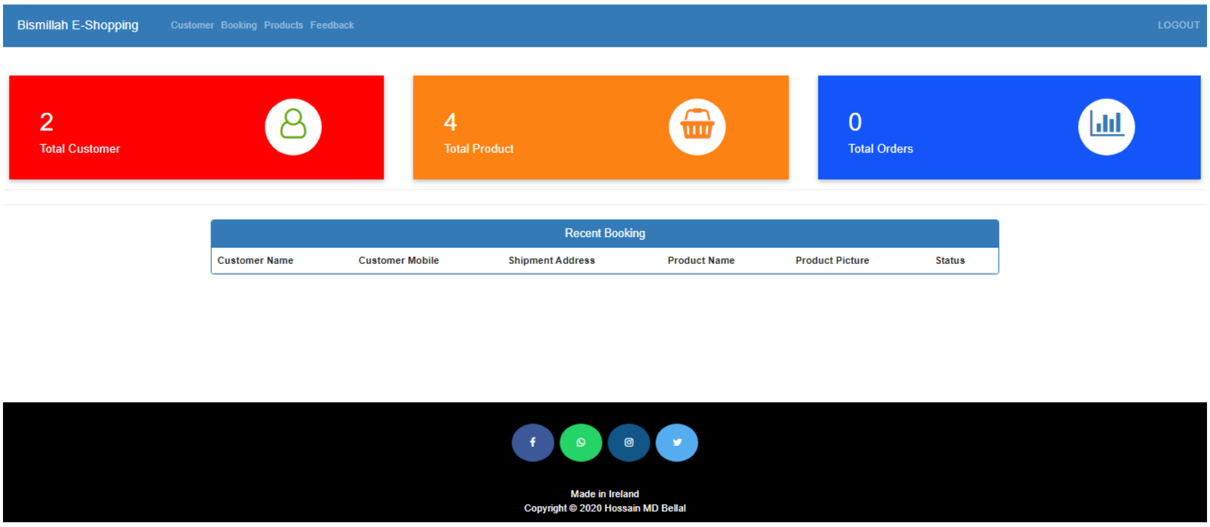
2.4.3 Login View

Once customer or admin has login credential, this page than to login to the application. If the user is an admin than it will lead the user to admin portal and if the user is a customer than the page will lead the user to customer homepage



2.4.4 Admin Dashboard View

This is the admin dashboard. Only authorised person can access this page. This page will appear to the admin after login the system. From this page admin has the option to control the product, maintain the sales(orders), customer information and also in this page, admin will see the product review (Feedback) from customer.



Bismillah E-Shopping

2
Total Customer

4
Total Product

0
Total Orders

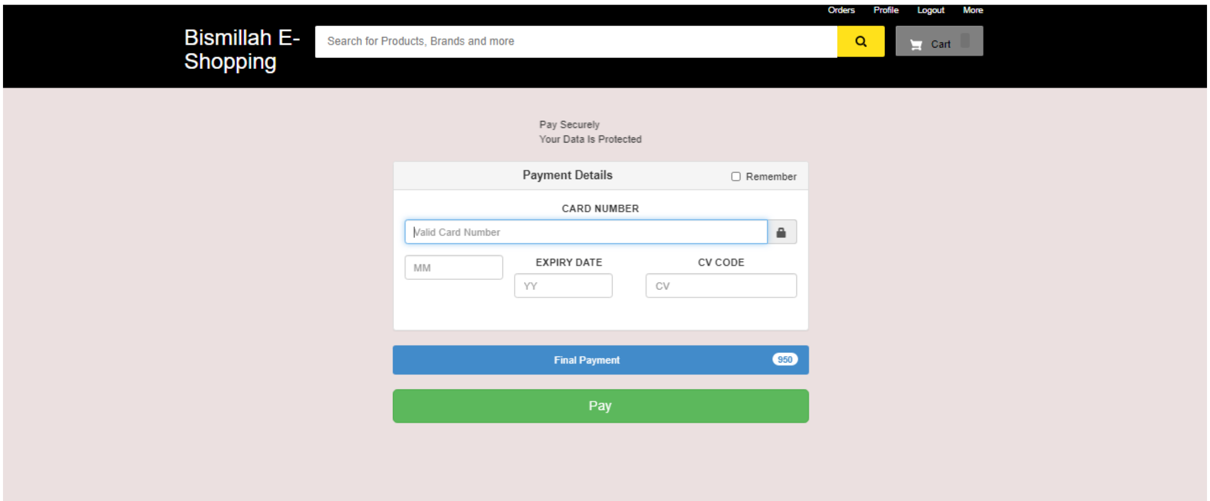
Recent Booking

Customer Name	Customer Mobile	Shipment Address	Product Name	Product Picture	Status

Made in Ireland
Copyright © 2020 Hossain MD Bellal

2.4.5 Payment option

This interface is to facilitate the purchase of the product. This page appreaed to customer after choosing the product and after determining the purchase. User must enter their relevant information to make the payment successful.



Bismillah E-Shopping

Orders Profile Logout More
Search for Products, Brands and more
Cart

Pay Securely
Your Data Is Protected

Payment Details

CARD NUMBER: Valid Card Number

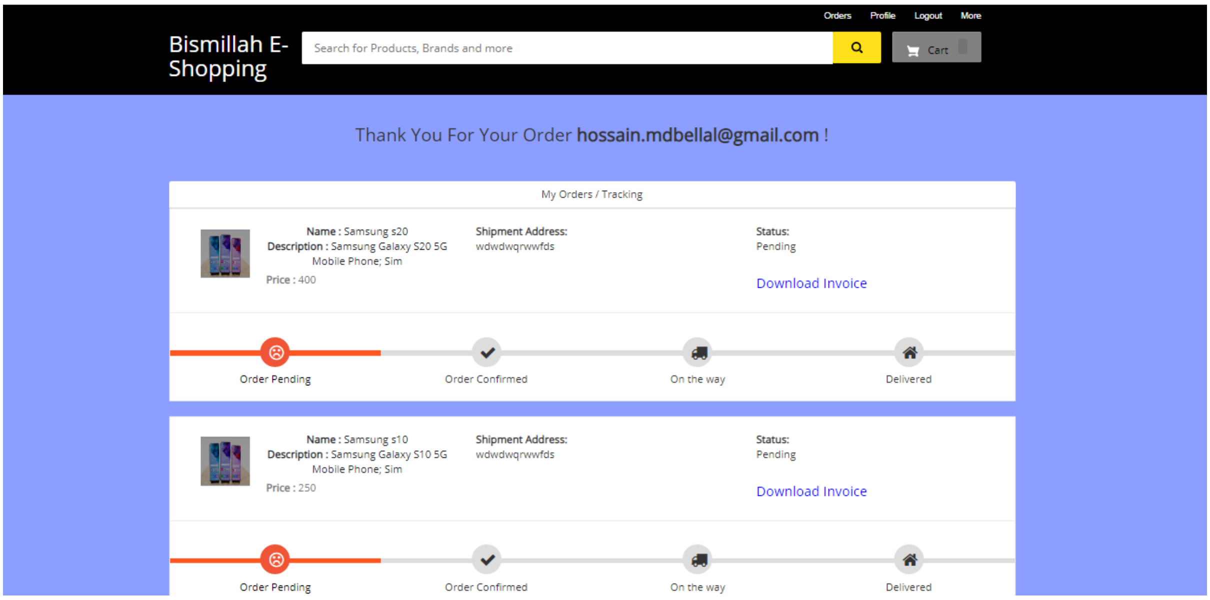
EXPIRY DATE: MM YY

CV CODE: CV

Final Payment 950

2.4.6 Order Details:

In this interface user will displayed their product information in details they have order and the status of the order can be checked here.



Thank You For Your Order **hossain.mdbellal@gmail.com** !

My Orders / Tracking

Name : Samsung s20
Description : Samsung Galaxy S20 5G Mobile Phone; Sim
Price : 400

Shipment Address:
wdwdwqrwwfds

Status:
Pending

Download Invoice

Name : Samsung s10
Description : Samsung Galaxy S10 5G Mobile Phone; Sim
Price : 250

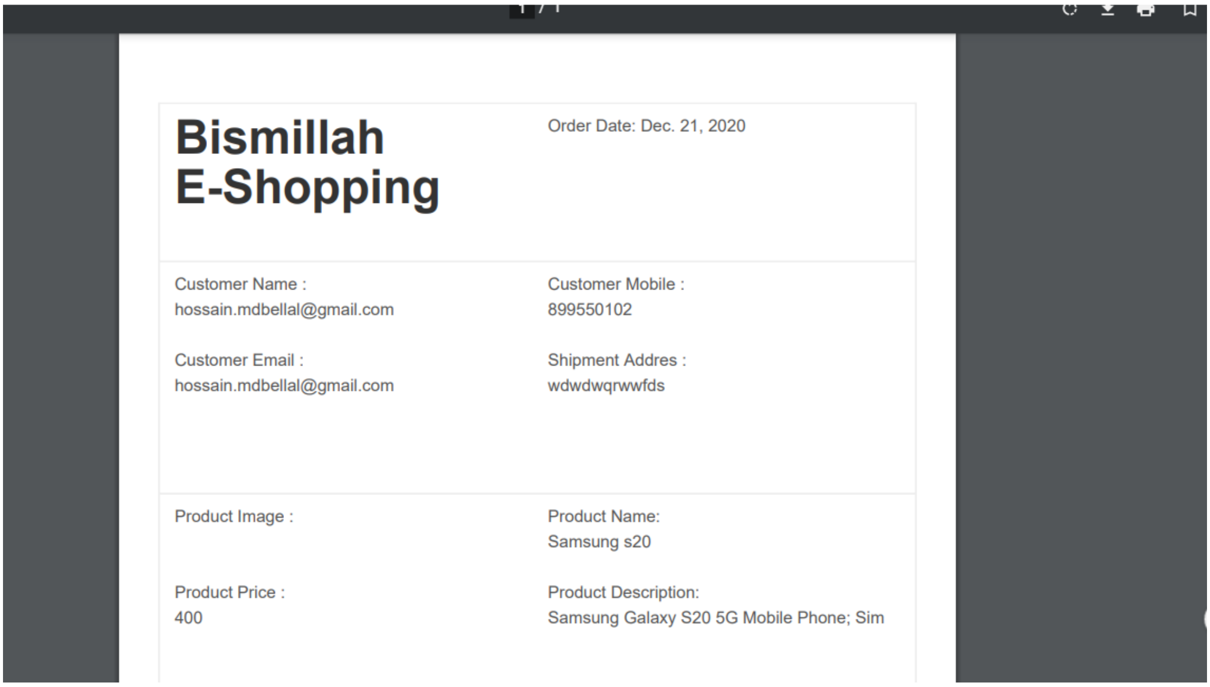
Shipment Address:
wdwdwqrwwfds

Status:
Pending

Download Invoice

2.4.7 order Invoice

This is the order invoice for customer. Customer can view that and also download that for their custom convenience



Bismillah E-Shopping

Order Date: Dec. 21, 2020

Customer Name :
hossain.mdbellal@gmail.com

Customer Mobile :
899550102

Customer Email :
hossain.mdbellal@gmail.com

Shipment Addres :
wdwdwqrwwfds

Product Image :

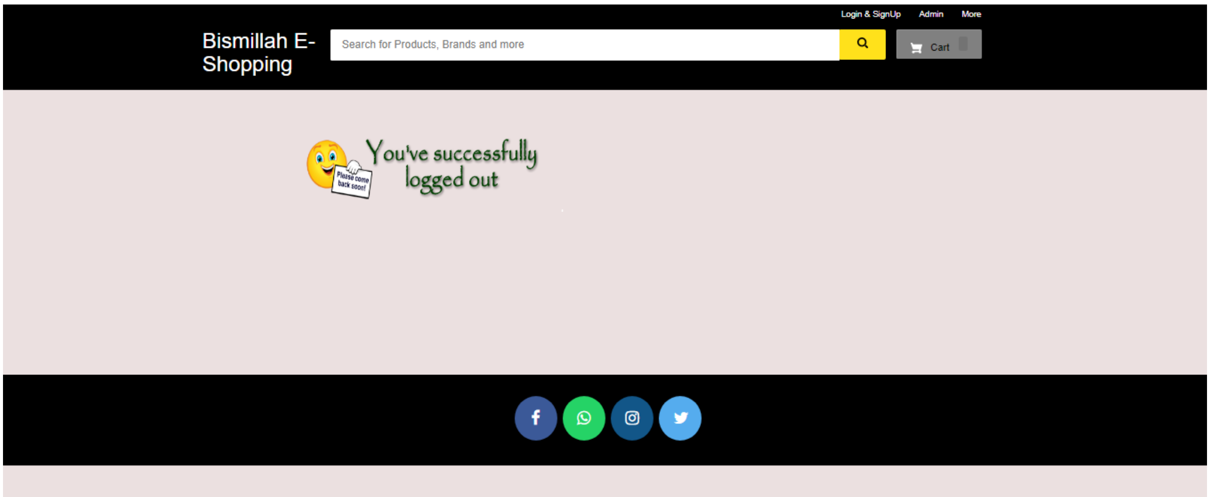
Product Name:
Samsung s20

Product Price :
400

Product Description:
Samsung Galaxy S20 5G Mobile Phone; Sim

2.4.8 Logout View

And finally this is the logout interface to logging out the system

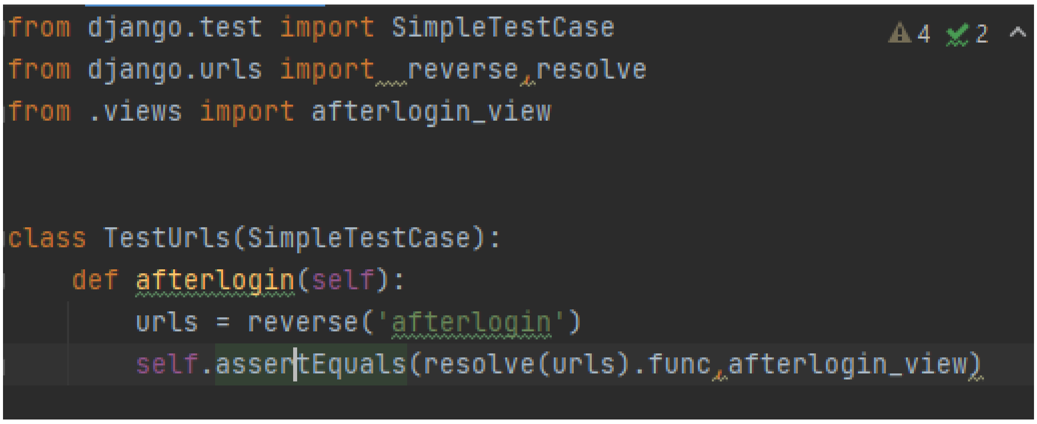


You've successfully logged out

2.5. Testing

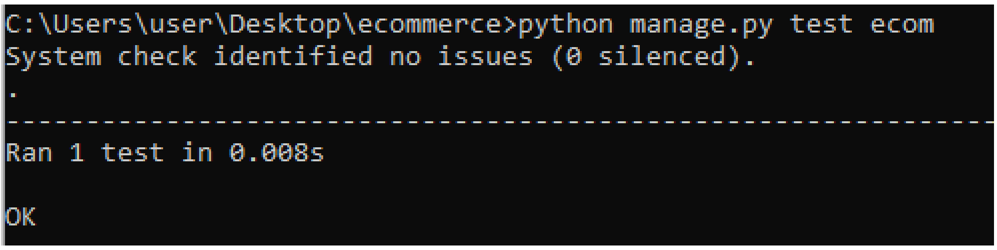
Unit Test

1. Urls test:



```
from django.test import SimpleTestCase
from django.urls import  reverse,resolve
from .views import afterlogin_view


class TestUrls(SimpleTestCase):
    def afterlogin(self):
        urls = reverse('afterlogin')
        self.assertEquals(resolve(urls).func,afterlogin_view)
```

Codebase for ‘afterlogin’ URLs test

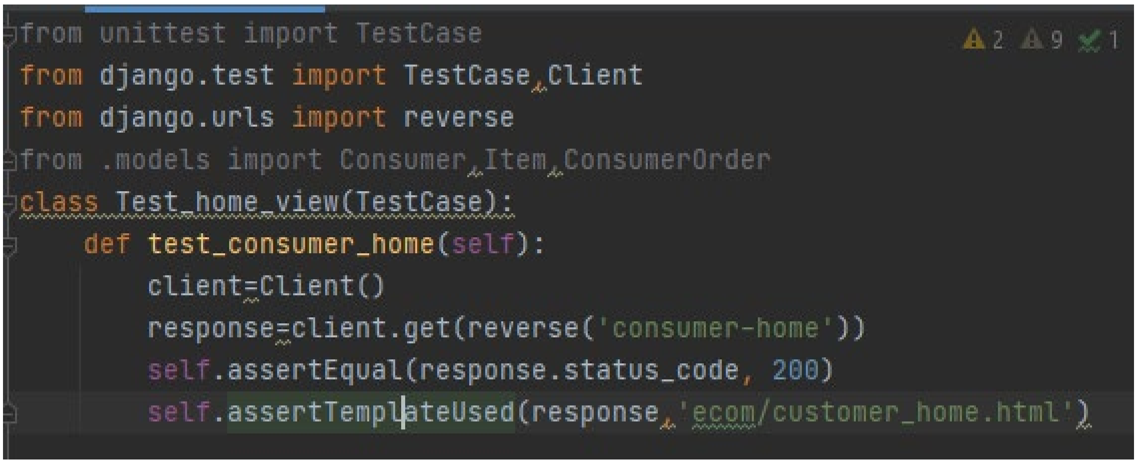


```
C:\Users\user\Desktop\ecommerce>python manage.py test ecom
System check identified no issues (0 silenced).
.
----------------------------------------------------------------
Ran 1 test in 0.008s

OK
```

Result of the testing

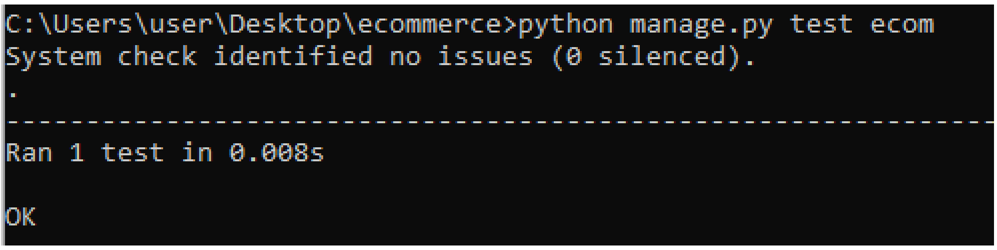
Result: from the above image we can see there is no error found for mentioned URLs and when run the application we find the expected result

View Test:



```
from unittest import TestCase
from django.test import TestCase,Client
from django.urls import reverse
from .models import Consumer,Item,ConsumerOrder
class Test_home_view(TestCase):
    def test_consumer_home(self):
        client=Client()
        response=client.get(reverse('consumer-home'))
        self.assertEqual(response.status_code, 200)
        self.assertTemplateUsed(response,'ecom/customer_home.html')
```

Codebase for testing ‘consumer_home’ view function

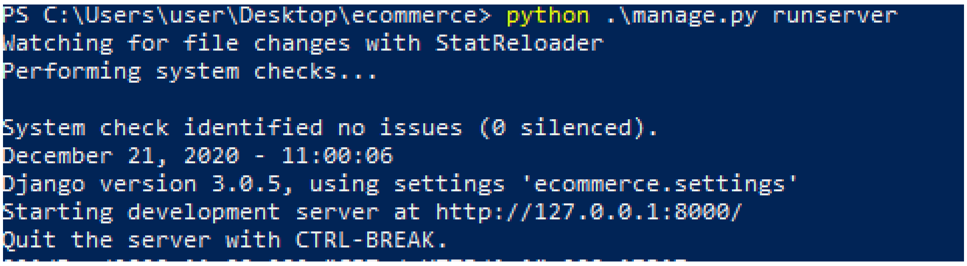


```
C:\Users\user\Desktop\ecommerce>python manage.py test ecom
System check identified no issues (0 silenced).
.
----------------------------------------------------------------------
Ran 1 test in 0.008s

OK
```

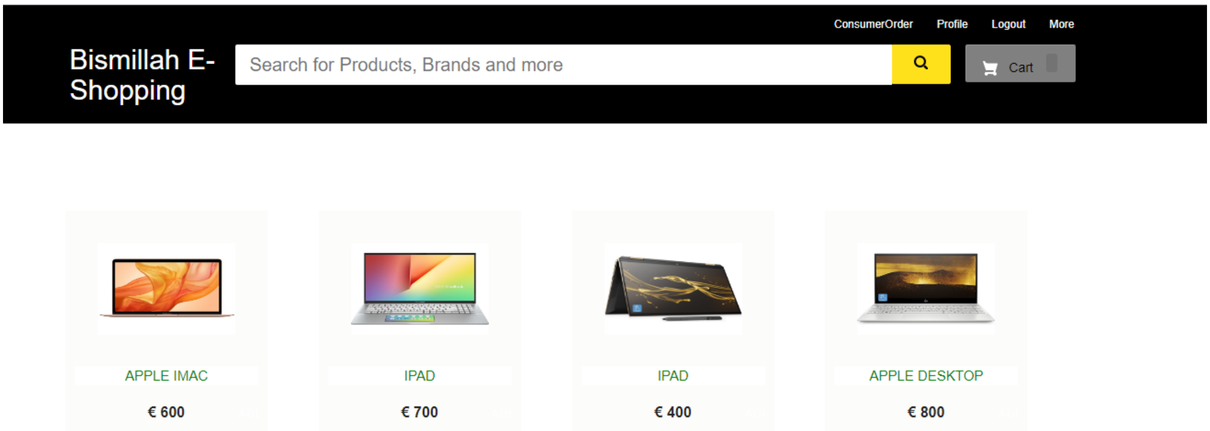
Console test result

Result: the view function is working as expected as there is no error

Local Host Testing for whole project:



```
PS C:\Users\user\Desktop\ecommerce> python .\manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 21, 2020 - 11:00:06
Django version 3.0.5, using settings 'ecommerce.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



And the above image shows the home-page of the application after testing the application on local host

2.6. Evaluation

A analysis of the site's user interface revealed the following questions and complaints about how the site functions with users who use it.

The Homepage:

A strong indicator of a business webpage is when a tourist can see what the company does and what can be accomplished on their website at a glance.

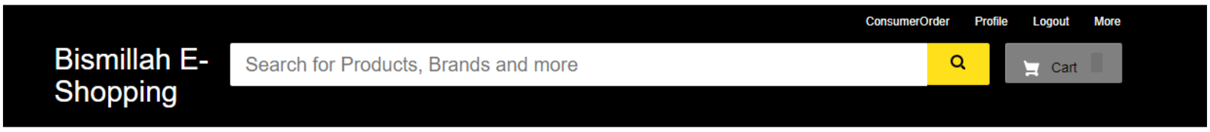
When the customer arrive at my applications homepage, it is pretty obvious what the web is about and what the organization does.

Online Ordering:

In Bismillah E-commerce application we have designed the application such way that anyone with even little knowledge of internet can make a order for their chosen products

Navigation Bar:

The key site navigation is well placed and simple to find and use. After all, it is not standardized around the web and varies abruptly and randomly as one navigates to the order fuel sections. This may be very frustrating for users of the web which results in a bad user experience. On the majority of the pages, the menu looks like bellow



After reviewing above all point we can overall marks the application as a moderate web-application.

3.0 Conclusions

Advantage:

Since it is an online shopping platform, it is easier to maintain the product control. Our Systems for eCommerce offer merchants the ability to provide registered customers with customized product and brand suggestions. By displaying the most important content to each consumer, these focused communications will help improve conversion.

Since my application has built on Django framework it protects against XSS attacks.

Disadvantage: Shoddily constructed technology paired with Python, which is not the quickest language in the world, will lead to slow websites. hence the user might get slow response when browse the application.

4.0 Further Development or Research

I have a long-terms plan with this application to further development to make it fulfil the client satisfaction. The further continues development and research will go on to add the new features such as live product analysis, product revies system, the product on sale page etc.

5.0 References

1. https://django-oscar.readthedocs.io/en/3.0.0/
2. https://github.com/Code-Institute-Submissions/django-ecommerce-1

6.0 Appendices

6.1. Project Proposal

Objectives

1. To Deliver a clean, responsive, restful web application within low cost
2. Provide a better, dynamic modern web application to boost client's business

Background

In current world situation people are more likely to shopping online rather than go to store physically. Some of the country's government discourage people not to go outside and to maintain their important activity online when sitting down at home. So online shopping becomes more attractive and the online market is larger than ever.

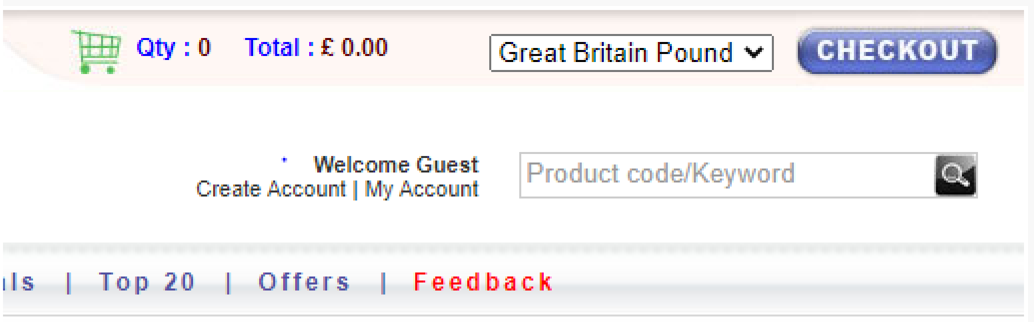
I am making a restful e-commerce web application called 'Bismillah e-Shopping' as my final year project. In my project we focused on to deliver a responsive, functional web application which will accede client expectations.

In our shop our target audience is anyone with small or medium size of business. The application will display various consumer product that business owner wants to sell on their shop with a responsive user interface. Any business entrepreneur can use my application for their online business with placing their own data. The application will have the backend server from where the management can maintain their product information. The user can view an item, rate the item, search the items, and buy the item with their PayPal account or visa debit/credit card. After complete purchase user will get confirmation by email and information about the item delivery. The user also can-do live chatting with the administration for any product information.

Technical Approach

Research:

I had a great research before I start my project as I do not want to be confused in any point after starting my project. I have discovered some advanced functionality that could be added on my project such as advance search function, ratting the displayed product, product promotion option etc.



Qty : 0 Total : £ 0.00

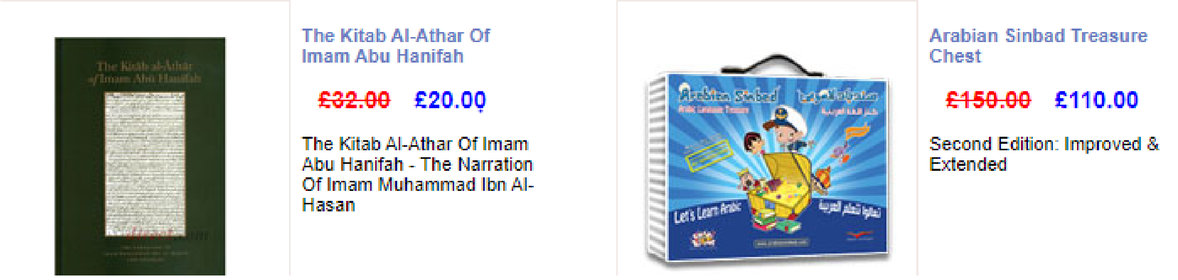
Great Britain Pound

CHECKOUT

Welcome Guest
Create Account | My Account

Product code/Keyword

ıls | Top 20 | Offers | Feedback



The Kitab Al-Athar Of Imam Abu Hanifah

~~£32.00~~ £20.00

The Kitab Al-Athar Of Imam Abu Hanifah - The Narration Of Imam Muhammad Ibn Al-Hasan

Arabian Sinbad Treasure Chest

~~£150.00~~ £110.00

Second Edition: Improved & Extended

Interface requirements:

1. User Interface (Application Interface)
2. User confidence
3. Keeping the user from unwanted purchase
4. Security of authorization information
5. Interface efficiency
6. Security to maintain user information
7. Froud prevention measure

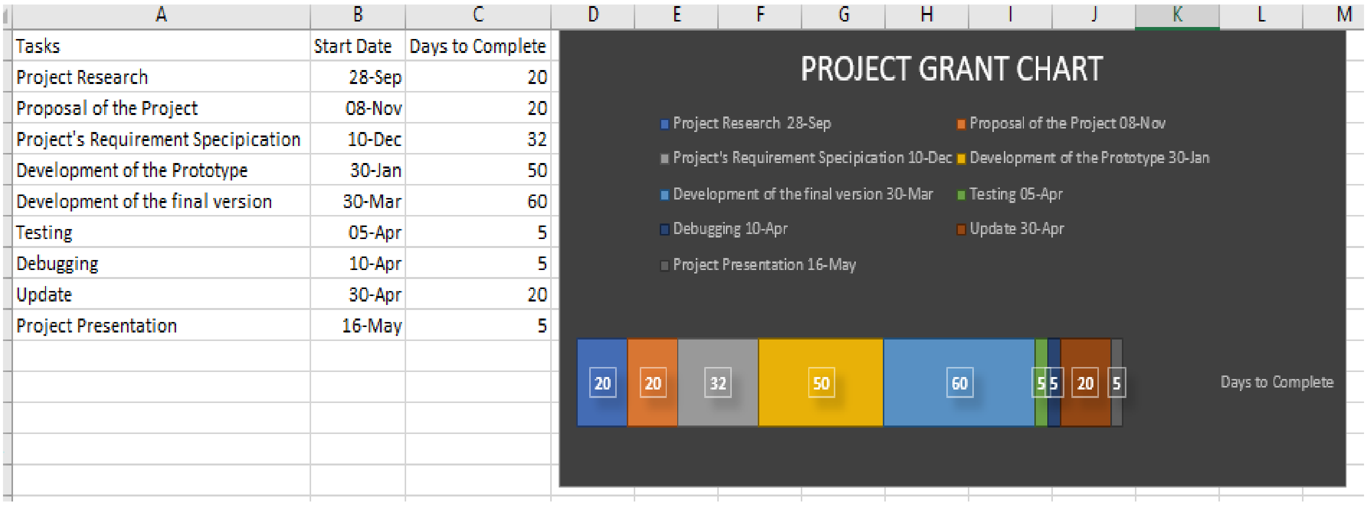
Implementation:

1. Build the initial project on local server
2. Create Backend interface
3. Create user interface
4. Test the application on local server
5. Deploy the application on web

Special Resources Required

1. Payment Gateway to execute the purchase such as PayPal or Strips
2. Domain name to deploy the application
3. Email host to send the e-mail

Project Plan



Tasks	Start Date	Days to Complete
Project Research	28-Sep	20
Proposal of the Project	08-Nov	20
Project's Requirement Specipication	10-Dec	32
Development of the Prototype	30-Jan	50
Development of the final version	30-Mar	60
Testing	05-Apr	5
Debugging	10-Apr	5
Update	30-Apr	20
Project Presentation	16-May	5

A
B
C
D
E
F
G
H
I
J
K
L
M
PROJECT GRANT CHART
Project Research 28-Sep
Proposal of the Project 08-Nov
Project's Requirement Specipication 10-Dec
Development of the Prototype 30-Jan
Development of the final version 30-Mar
Testing 05-Apr
Debugging 10-Apr
Update 30-Apr
Project Presentation 16-May
20
20
32
50
60
5
5
20
5
Days to Complete

Technical Details

Language:

1. Python: I will use Python 3.7 programming language for my project as backend language. Python is a very powerful language to work fast and to integrate the system quickly.
2. Java Script: I will use java script programming language to add interactive behaviours to my user interface
3. HTML: I also will use HTML 5 to design my user interface
4. CSS: for styling my interface I will user CSS stylesheet

Library: I will use python channel to implement live chatting system

Framework: There is a framework comes with python called Django. I will use Django 2.1 to build my application

Editor: I will use “Pycharm” as editor

6.2 Reflective Journals 1

In my software project module, it is mandatory to build an individual project. I was confused about what kind of project could I make. I was doing some online research on my area of interest to find out something impressive. My area of interest is web development as I already work few projects with Python Django framework. but still I was in bit stress to choose my project.

After long research I have felt that I should make my project in my area of interest, and there should be a print of my course specialization as well. my thinking is clear enough now about what need to do next

Finally, I have decided to work to build an e-commerce project and i already submitted my project pitch video and waiting for approvement of it from my lecturer. My project proposal is due to submit next week and I am working on it now.