TSE2101
Final Report

for

Quotation System

Version <3.0>

Tutorial Section: TT2L

Group No.: *Group 4*

Ranjani A/P Tamil Chelvan	**1211303208**
Fawzul Bari Bin A A Naina Mohamed	**1211301743**
Srihari Naidu A/L Venkatash	**1201100182**
<name>	**<student #>**

Date: 26/1/2023

CONTENTS

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

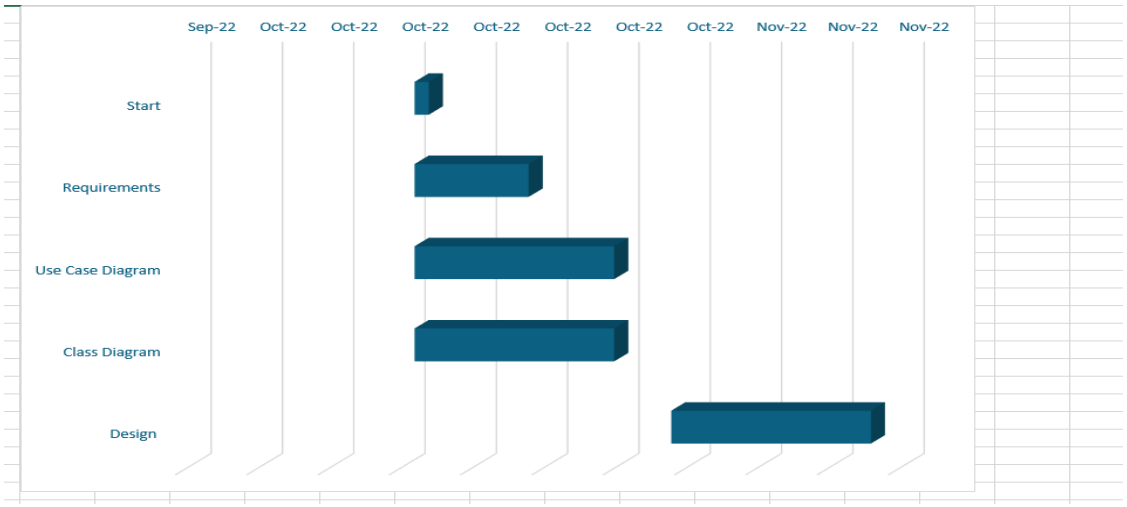
1 Project Management

1.1 Team Members

Name	Actor/Processes
Ranjani A/P Tamil Chelvan	Manager
Fawzul Bari Bin A A Naina Mohamed	Salesman
Srihari Naidu A/L Venkatash	Finance Officer

1.2 Project Plan

Gantt project chart1.ods **(Link to gantt chart)**



This gantt chart is to track our progress and ensure we are doing the assignment by the given dates.This gantt chart makes our progress more trackable and we get to finish our assignment task based on the dates that are planned in the gantt chart.

2 System Overview

2.1 Description

The product, Quotation System, aims to support a Vendor in a set scope. The said scope of this system is from receiving a Purchase Requisition to receiving a Purchase Order from a Customer. This derives 4 different users/actors in this system; Manager, Salesman, Finance Officer and Customer.

The flow of the system and its functionalities goes as follows:

1. All users of the system must be registered and logged in the system to perform any actions.
2. The Customer sends a Purchase Requisition form, containing a list of items they're interested in, to the Vendor.
3. The Salesman views the Purchase Requisition form, then generates a Quotation form containing the price details for the list of items, and sends it to the Manager.
4. The Manager checks the Quotation form, approves it and then it is sent to the Customer.
5. The Customer then views the Quotation and sends a Purchase Order form, containing the items to be purchased and its quantity, to the Vendor.
6. The Finance Officer then views the Purchase Order form.

Additional functionalities includes:

1. The Manager can view the purchase order report.

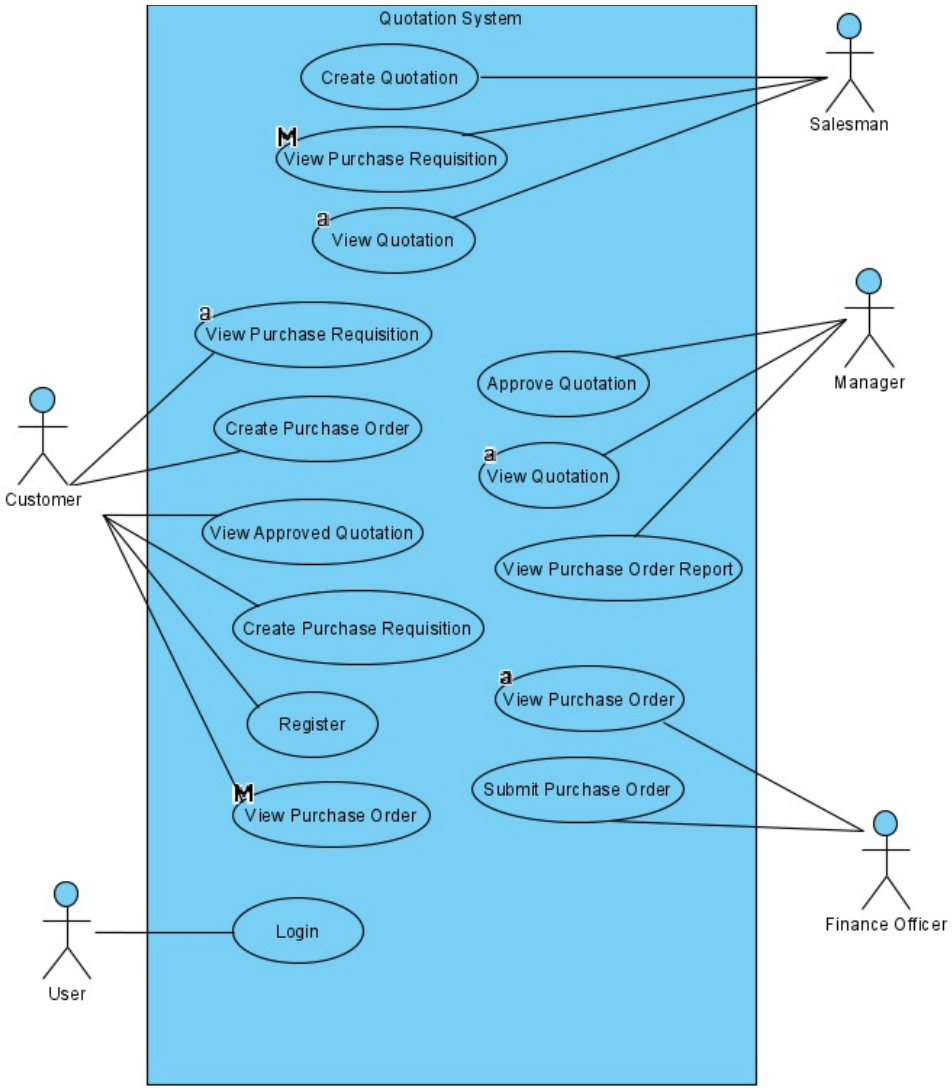
2.2 Actors

Actor	Use Cases
Manager	Login
	View Purchase Order Report
	Approve Quotation
	View Quotation
Salesman	Login
	View Purchase Requisition
	Create Quotation
	View Quotation
Finance Officer	Login
	Submit Purchase Order
	View Purchase Order
Customer	Login
	Register
	View Purchase Requisition
	Create Purchase Order
	View Purchase Order
	Create Purchase Requisition
	View Approved Quotation

2.3 Assumptions and Dependencies

- The Manager, Salesman and Finance Officer are pre-registered in the system.
- The Customer with multiple accounts under the same Company name will be treated as different customers.
- The Customer has to resend their Purchase Requisition if they made a mistake with the first one—and a Quotation still can be created for the wrong Purchase Requisition.

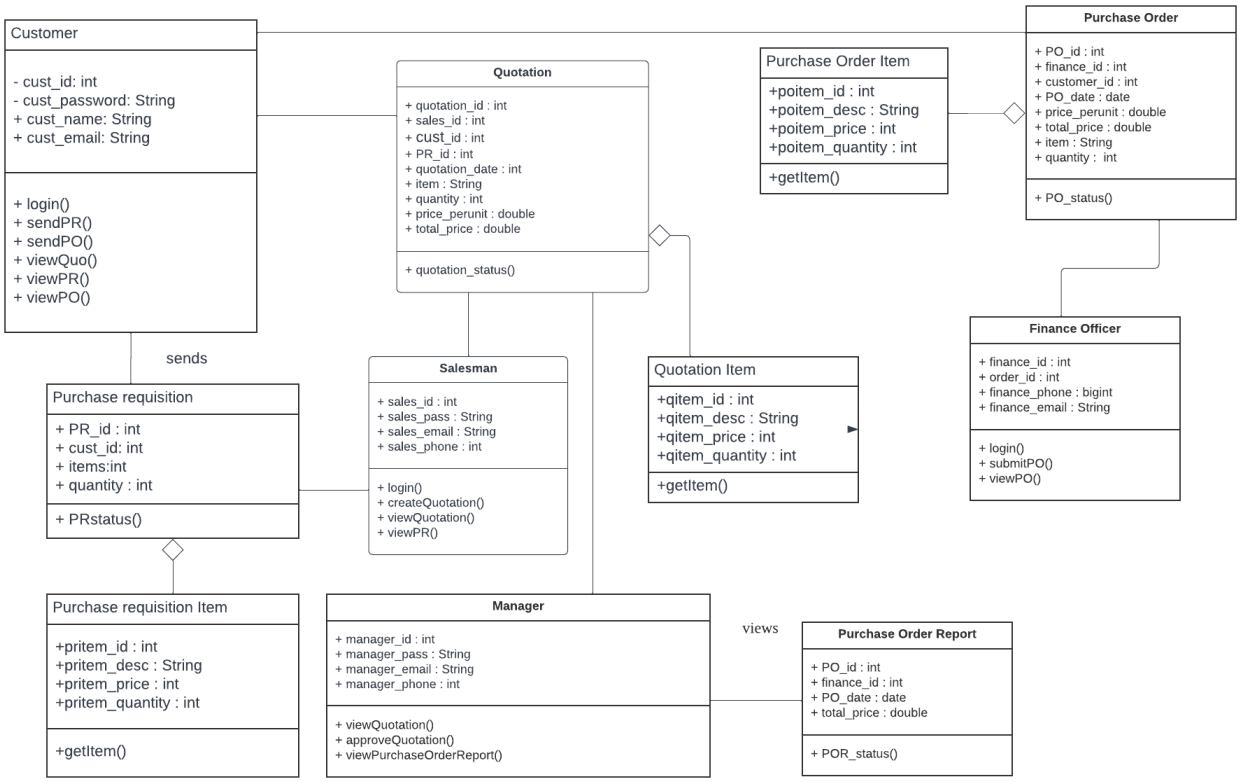
2.4 Use Case Diagram



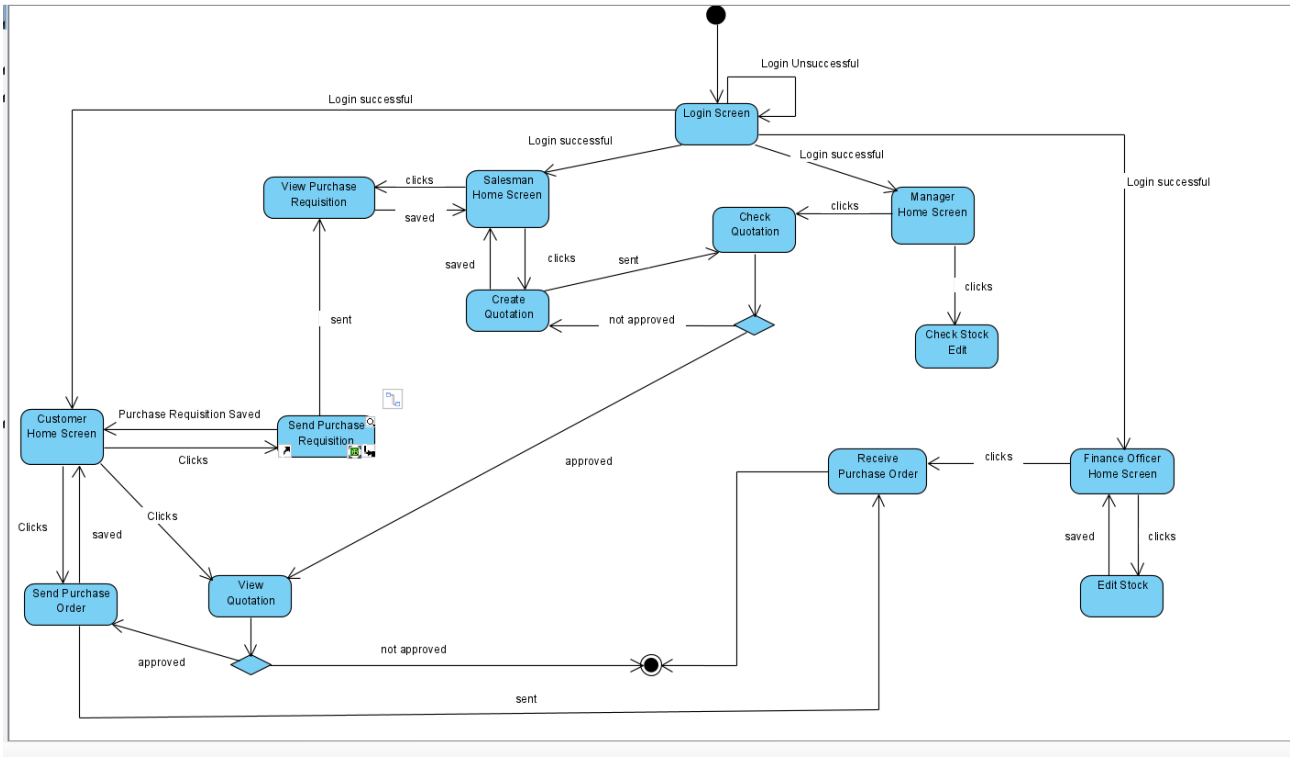
3 Requirements

3.1 Class Diagrams

This class diagram contains all the classes we are going to implement in this system. The system has 11 classes.

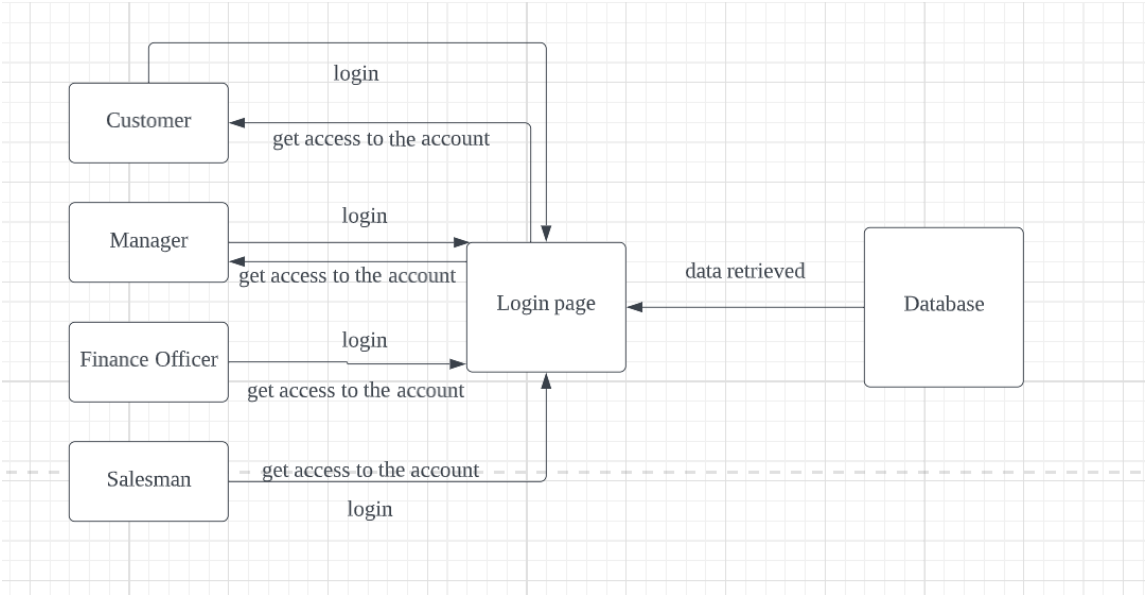


3.2 State Diagrams

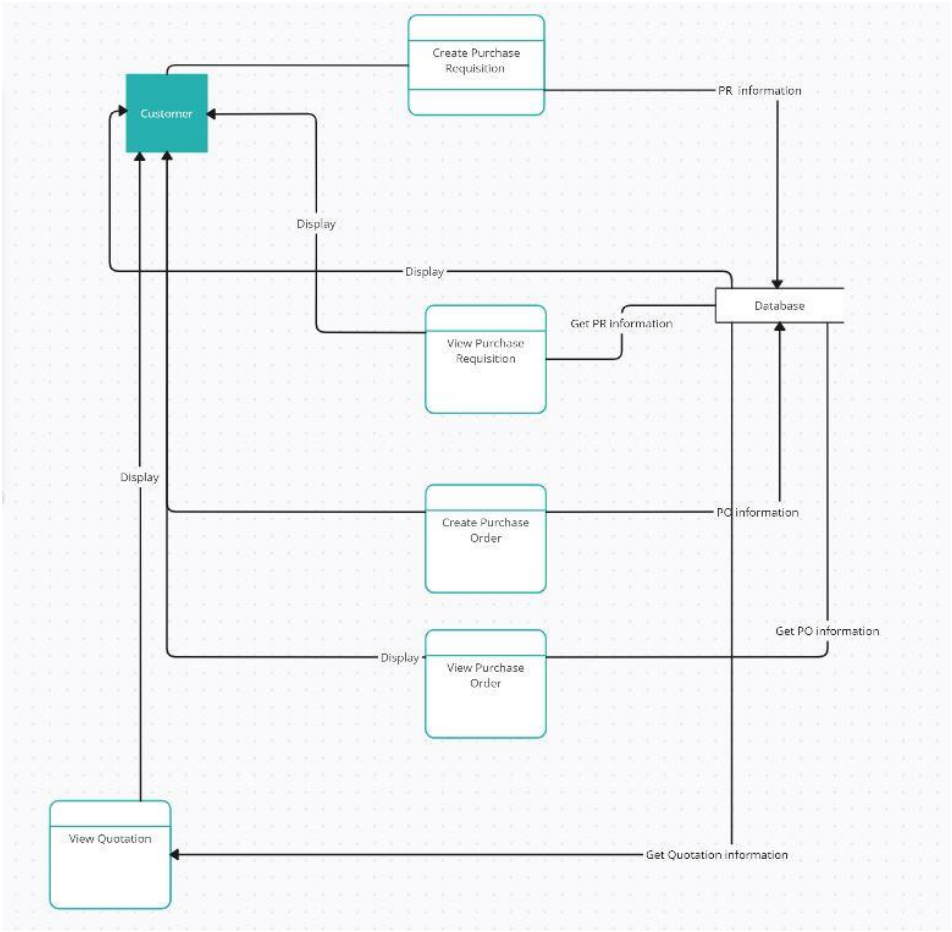


This state transition diagram contains all the possible transitions that might occur in this system based on the user's action. The login process will go on loop until the user enters the correct email and password into the system. Then it will be directed to a specific homepage based on the user, Customer, Manager, Finance Officer and Salesman. Each user has different states after that point. Finally the system will be terminated once the user logs out.

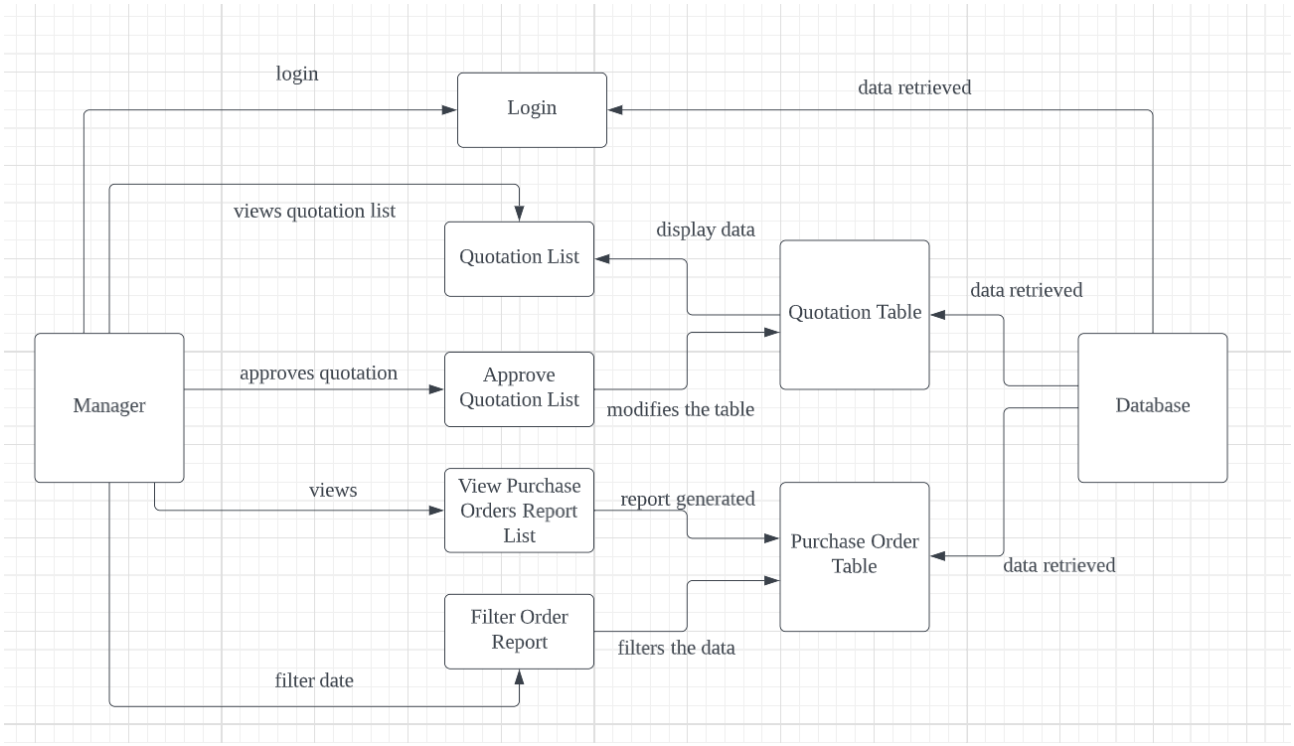
3.3 Data Flow Diagrams



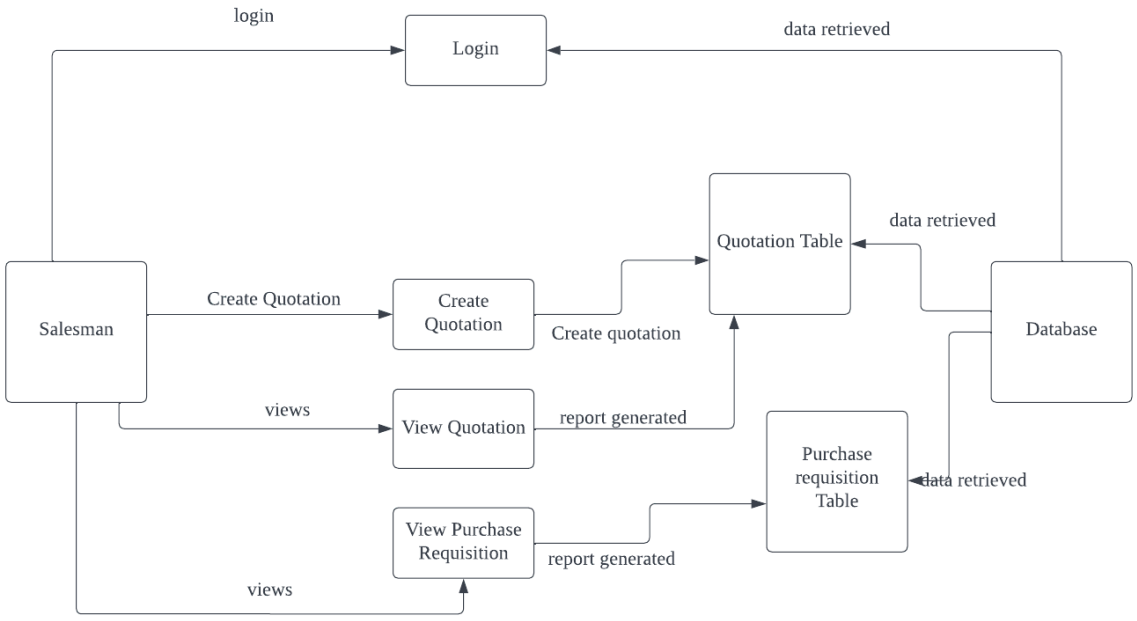
3.3.1 Customer Diagram



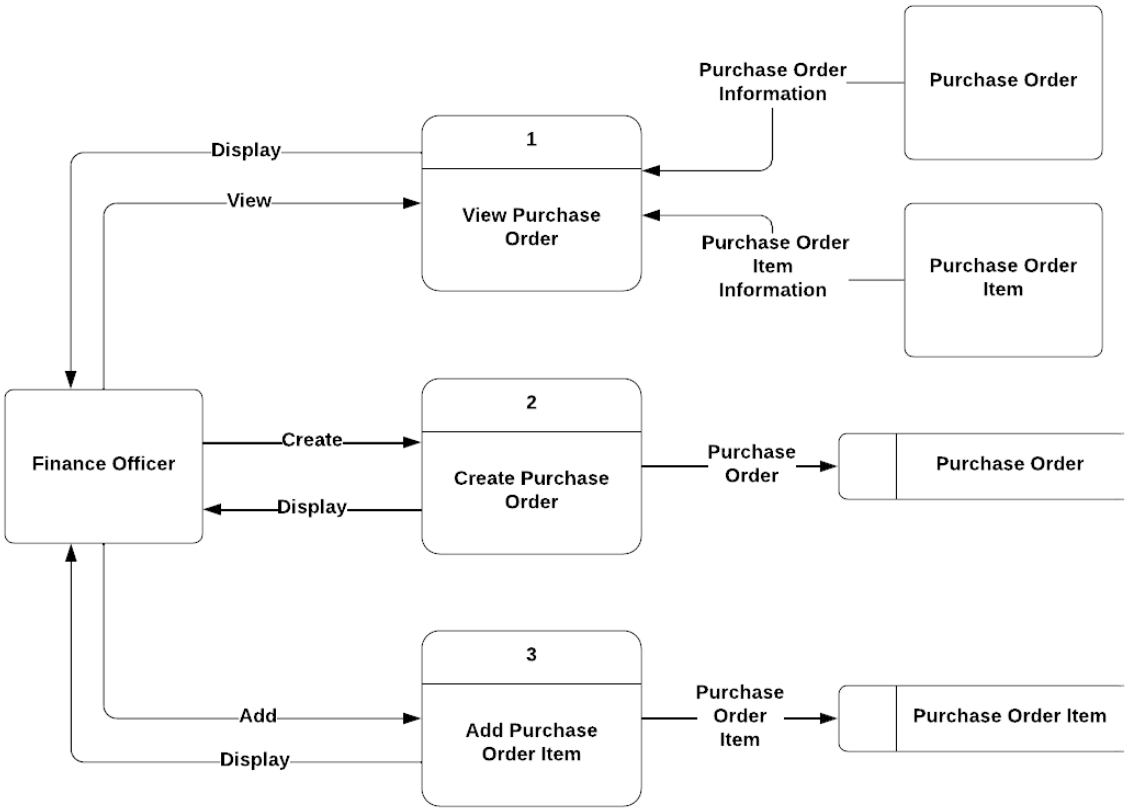
3.3.2 Manager Diagram



3.3.3 Salesman Diagram



3.3.3 Finance Officer Diagram



4 Design

4.1 Data Dictionary

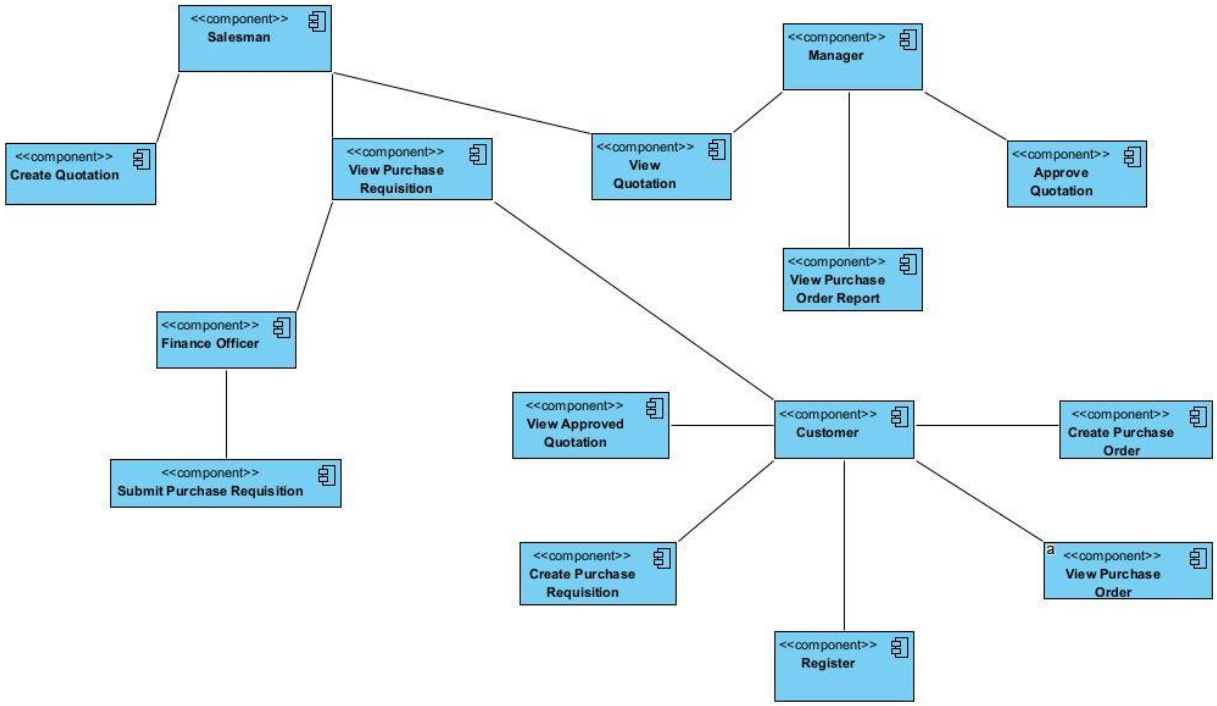
Table Name	Attribute Name	Description	Data Type	Required	Key
User Table	username	User identification number	Varchar(20)	Y	PK
	password	User password	Varchar(20)	Y	
	first_name	User firstname	Varchar(30)	Y	
	last_name	User lastname	Varchar(30)	Y	
	email_address	User email address	Varchar(50)	Y	

Purchase Order table	id	Purchase order id	int	Y	PK
	officer	Owner user	int	Y	FK
	date	Purchase Order date	date	Y	
	company_name	Customer's company name	Varchar(100	Y	
	address_line_1	Customer's Delivery address	Varchar(100	Y	
	address_line_2		Varchar(100)	Y	
	city		Varchar(50)	Y	
	state		Varchar(20)	Y	
	zipcode		Varchar(10)	Y	
	country		Varchar(20)	Y	
	contact_name	Customer's Contact number	Varchar(50)	Y	
	contact_no	Customer's Contact number	Varchar(20)	Y	
Purchase Requisition Table	PR_id	Purchase requisition id	int	Y	PK
	cust_id	Customer's identification number	int	Y	FK
	pritem_id	Requested items	Varchar(30)	Y	FK
	PR_date	Purchase requisition date	date	Y	
Quotation Table	quot_id	Quotation unique id	int	Y	PK
	salesstaff_id	Salesman unique id number	int	Y	FK
	cust_id	Customer unique id number	int	Y	FK
	PR_id	Purchase requisition id	int	Y	FK
	qitem_id	Quotation item	Varchar(50)	Y	FK
	quot_date	Quotation date	date	Y	

	quo_tprice	Quotation total price	int	Y	
Purchase Order Report	PO_id	Purchase order id	int	Y	PK, FK
	finance_id	Finance officer's identification number	int	Y	
	PO_date	Ordered date	date	Y	
	PO_tprice	Ordered item total price	int	Y	
Purchase Requisition Items	pritem_id	item identification number	int	Y	PK
	pritem_desc	item description	Varchar(100)	Y	
	pritem_quantity	item quantity	int	Y	
	pritem_price	item price	int	Y	
Purchase Order Items	id	Purchase Order Item ID	int	Y	PK
	po	Owning Purchase Order	int	Y	FK
	name	Item's name	Varchar(100	Y	
	price	item price	int	Y	
	quantity	item quantity	int	Y	
Quotation Items	qitem_id	item identification number	int	Y	PK
	qitem_desc	item description	Varchar(100)	Y	
	qitem_quantity	item quantity	int	Y	
	qitem_price	item price	int	Y	

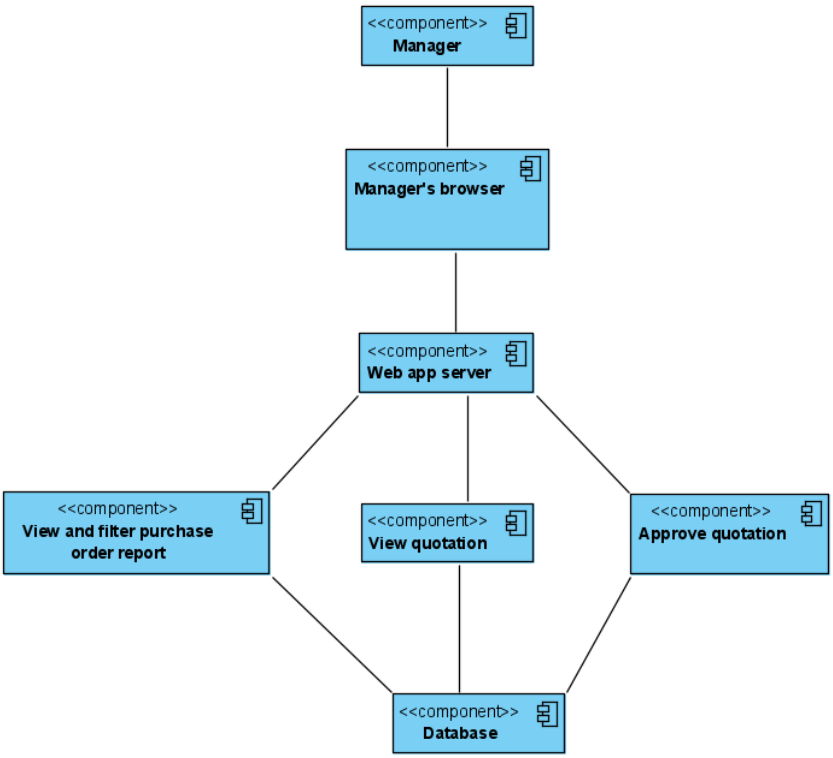
4.2 Data Structures

4.3 Software Architecture



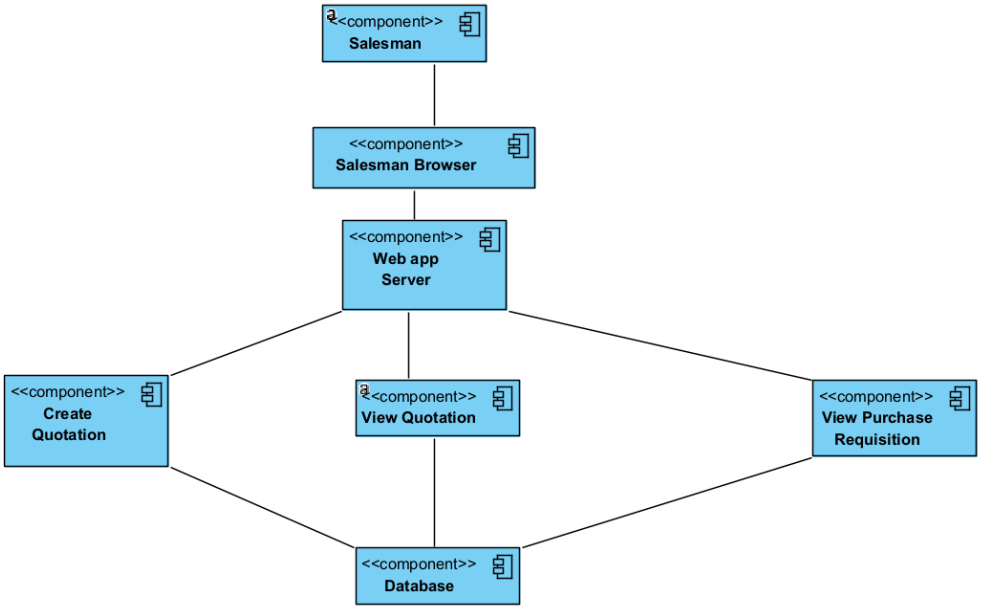
The above diagram shows the architecture of our system. Each actor has different access to different parts of the system, some of the features of the system are accessible to multiple actors and are able to view certain documents that are recorded in the system.

4.3.1 Manager Subsystem



4.3.2 Customer Subsystem

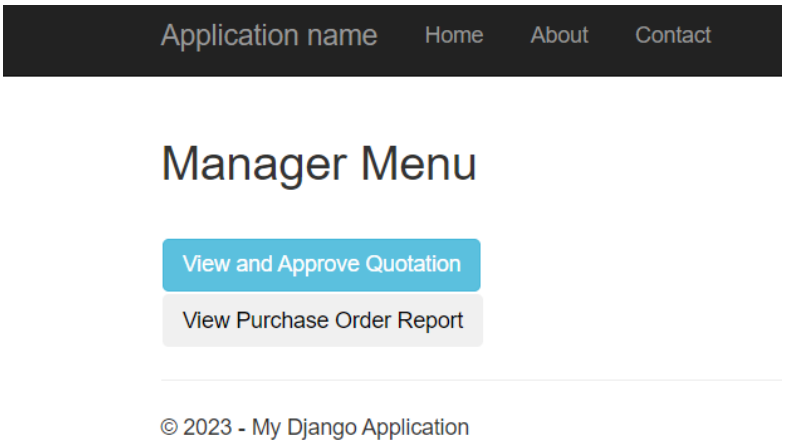
4.3.3 Salesman Subsystem



4.3.4 Finance Officer Subsystem

4.4 Main Screens

Manager Home Screen



Purchase Order Report view screen

Purchase Orders List

Detailed View

© - My Django Application

PO_id	finance_id	PO_date	PO_tprice
11	22	Jan. 9, 2023	33
12	23	Feb. 2, 2023	155
44	20	March 22, 2023	77
77	49	April 20, 2023	4134

Select time period

Purchase Order Report

From : dd/mm/yyyy To : dd/mm/yyyy Search

© - My Django Application

PO_id	finance_id	PO_date	PO_tprice
11	22	Jan. 9, 2023	33
12	23	Feb. 2, 2023	155

Quotation List

© - My Django Application

quote_id	salesstaff_id	cust_id	pr_id	qitem_id	quot_date	quo_tprice	status	Update
4	5	6	3	23	Jan. 17, 2023	67	pending	approve
7	4	2	5	88	Jan. 9, 2023	33	pending	approve

quote_id

345

salesstaff_id

3445

cust_id

3454

pr_id

3345

qitem

11

quot_date

Jan. 13, 2023

quo_tprice

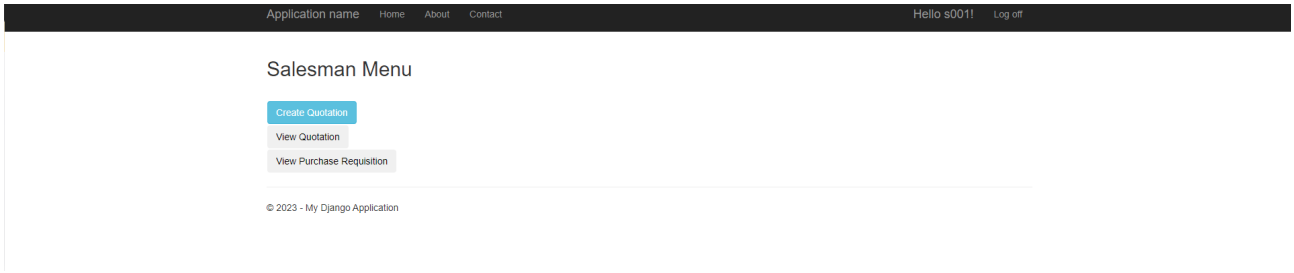
6767

status

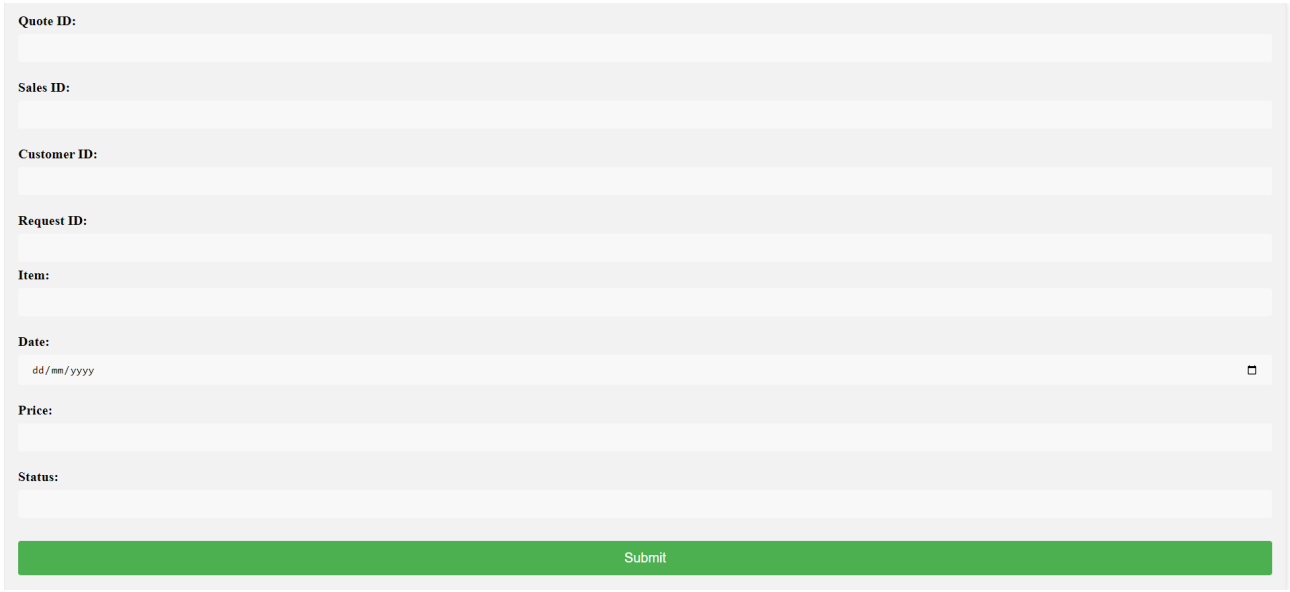
approved

Update

Salesman Home Screen



Create Quotation Screen



View Quotation Screen

Quotation Table

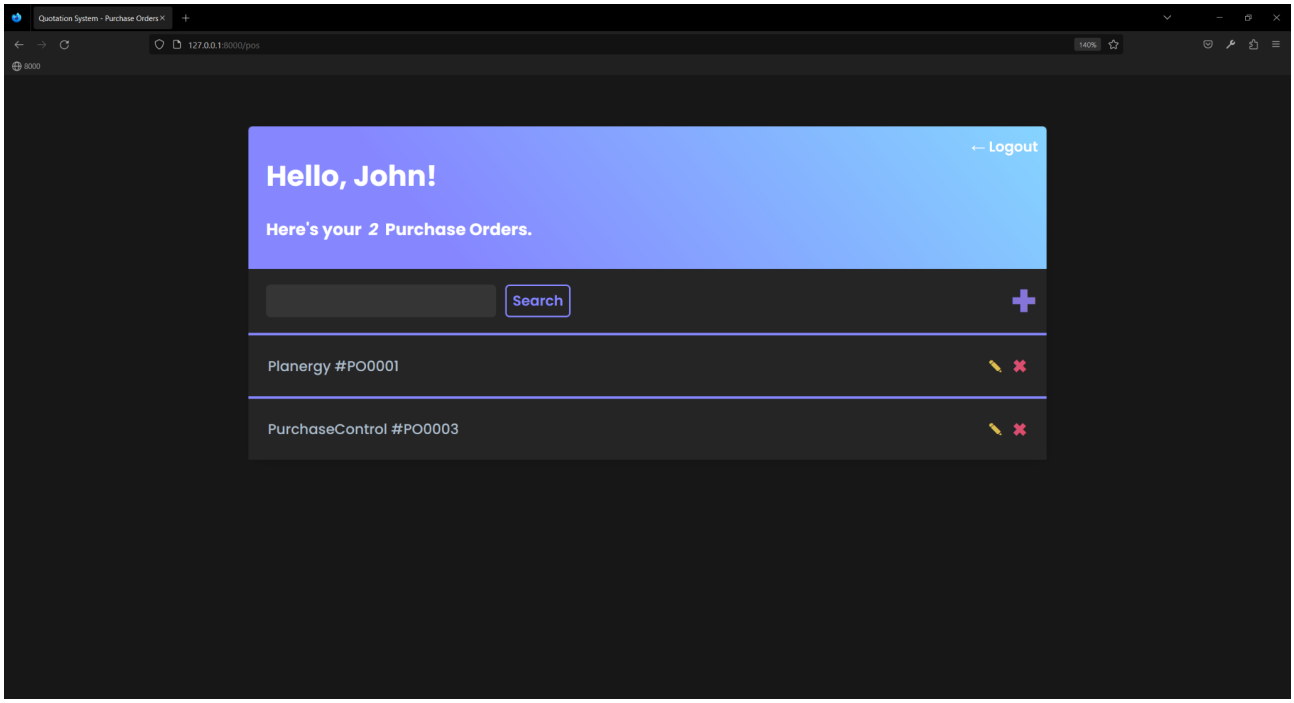
Quote ID	Sales ID	Customer ID	Request ID	Item	Date	Price	Status
4444	12345	1111	dell	2222	Jan. 18, 2023	1000	pending
4444	1111	2222	1111	22222	Jan. 18, 2023	222	pending
4444	3444	134521	34555	dell	Jan. 14, 2023	333	pending
1111	2222	3333	4444	Tablet	Jan. 19, 2023	300	pending
1211	2233	1123	11233	13331	Jan. 18, 2023	23411	sent

View Purchase Requisition Screen

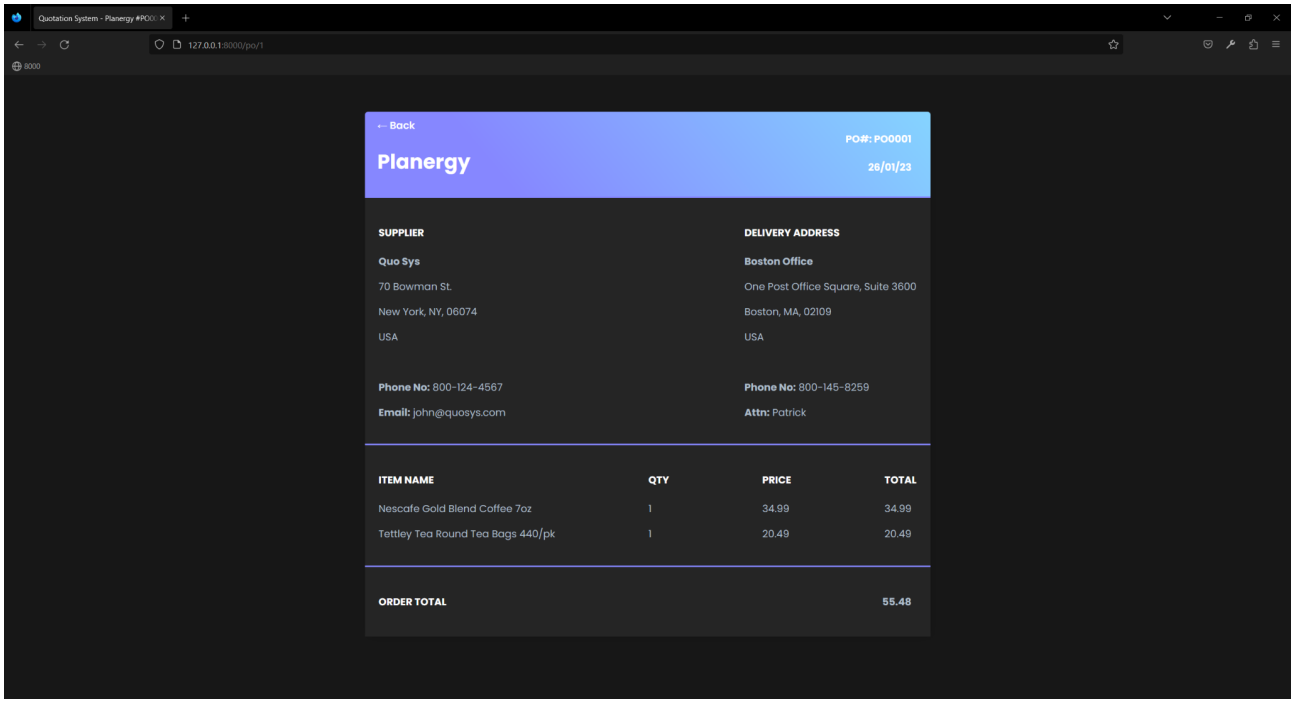
View Purchase Requisition

PR ID	Customer ID	PR Item ID	PR Date
23	45	77	Jan. 2, 2023
45	23	11	Jan. 4, 2023

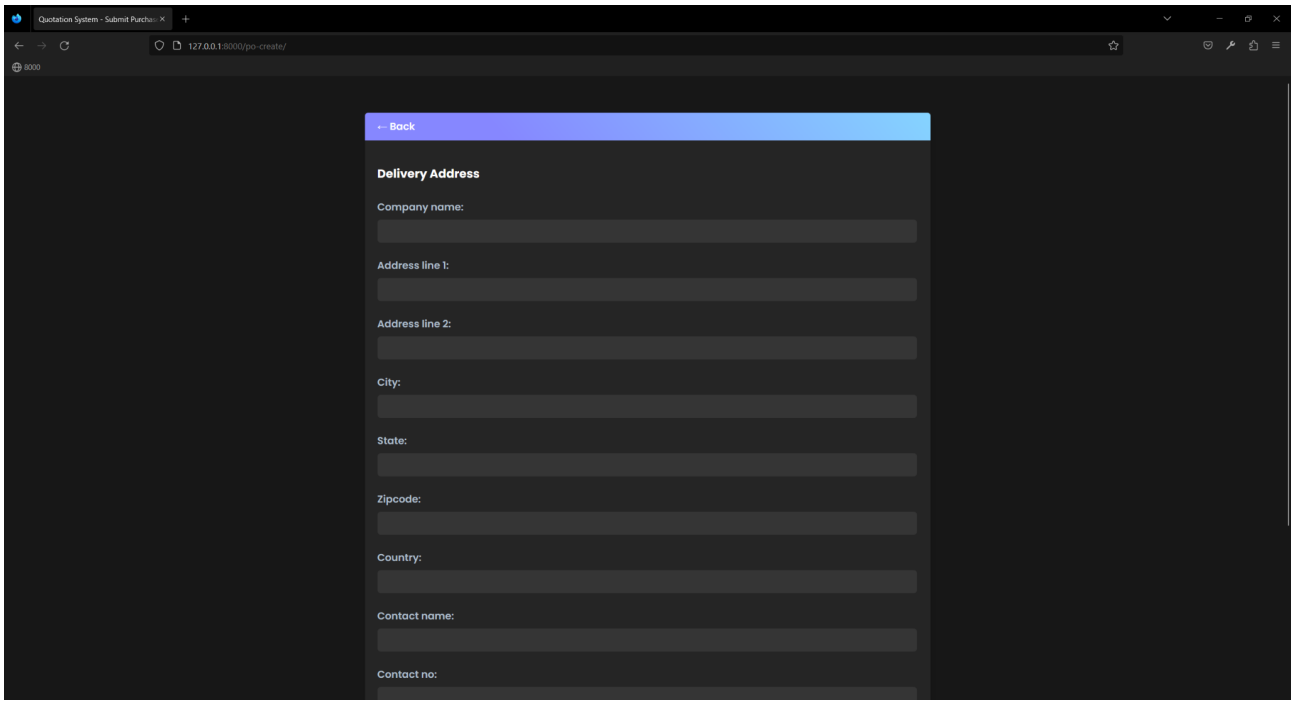
View Purchase Order List

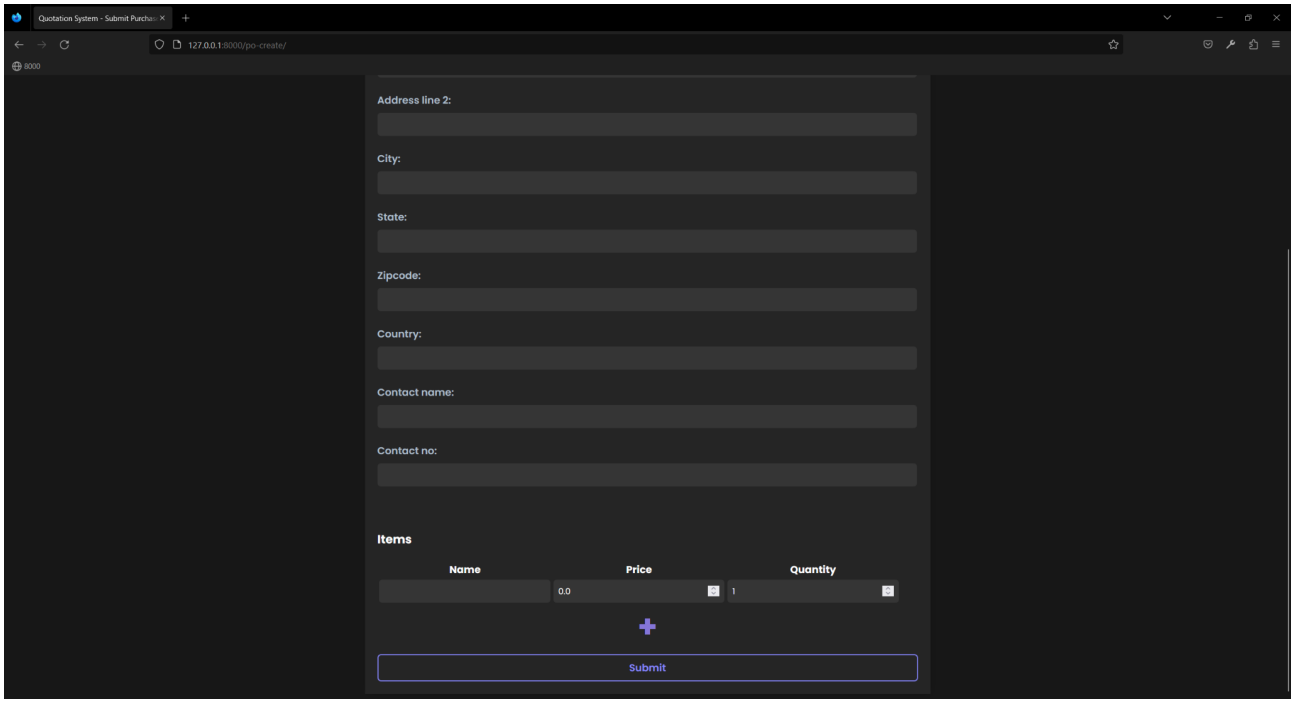


View Purchase Order

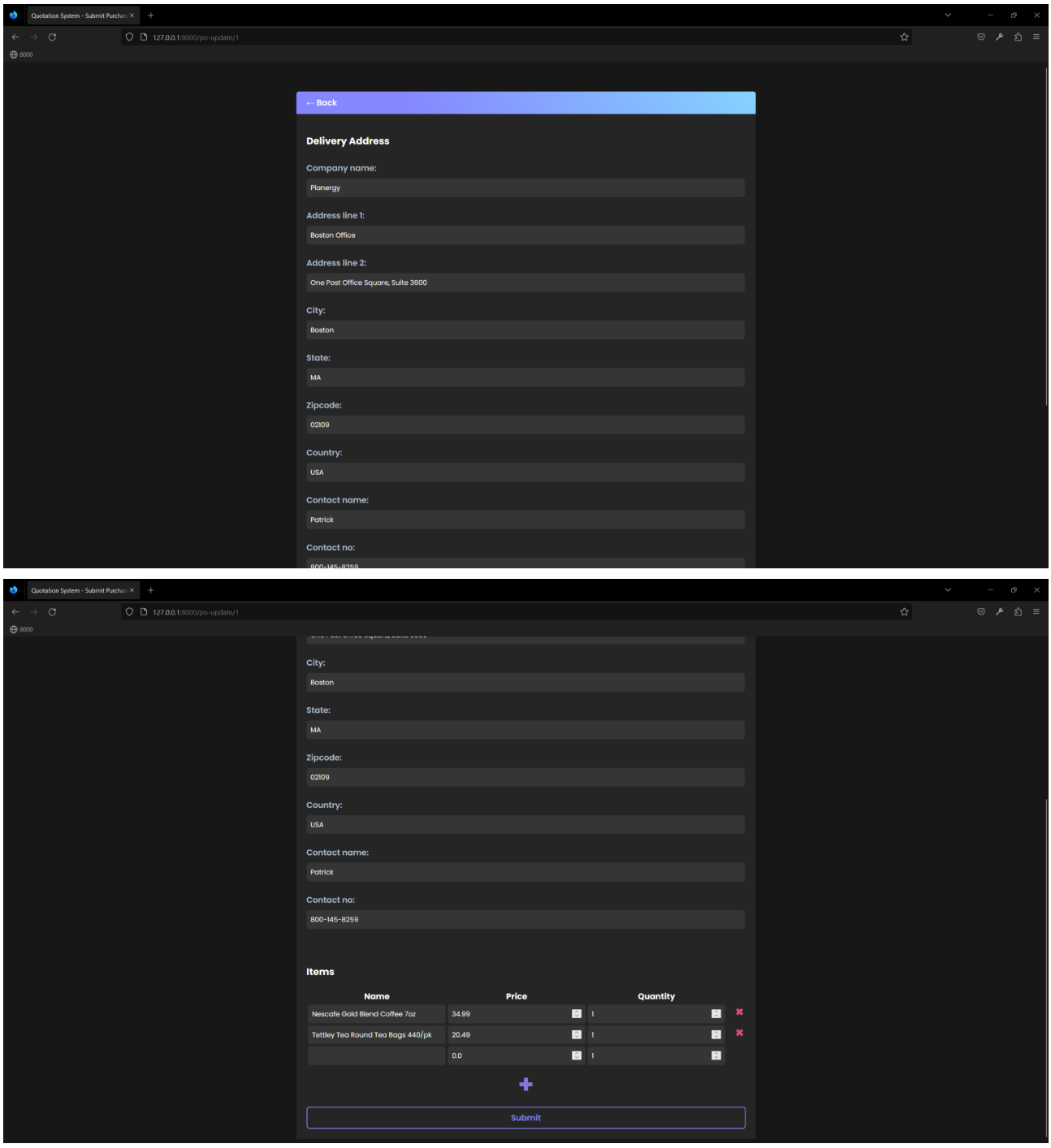


Create Purchase Order

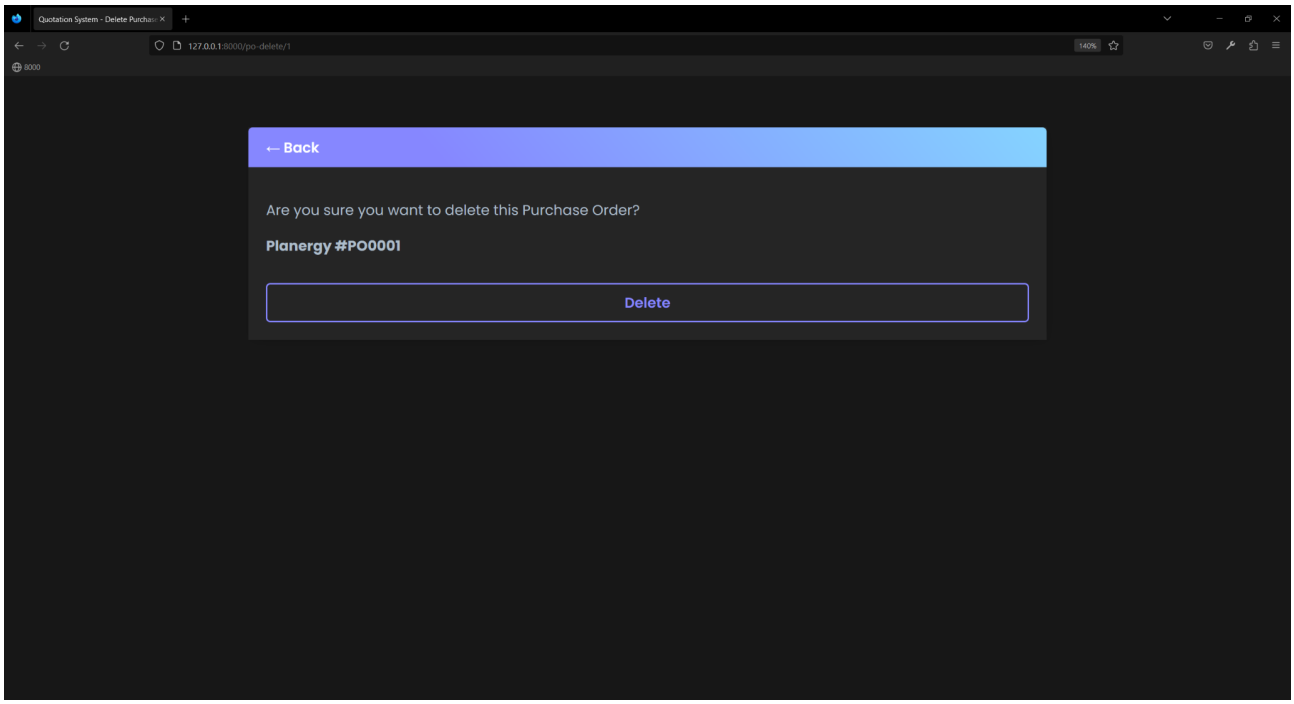




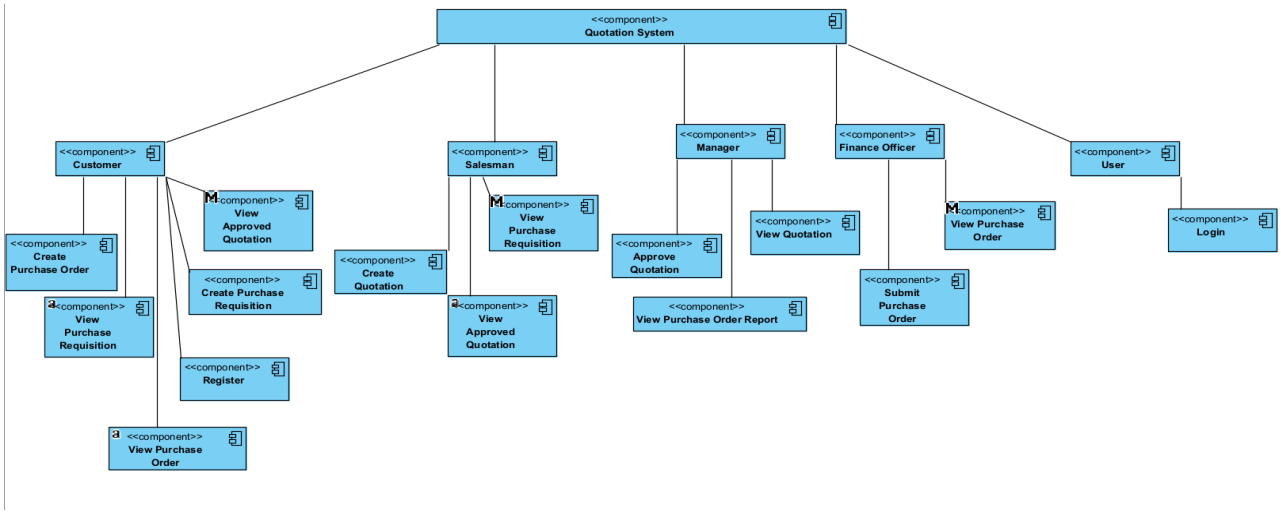
Edit Purchase Order



Delete Purchase Order

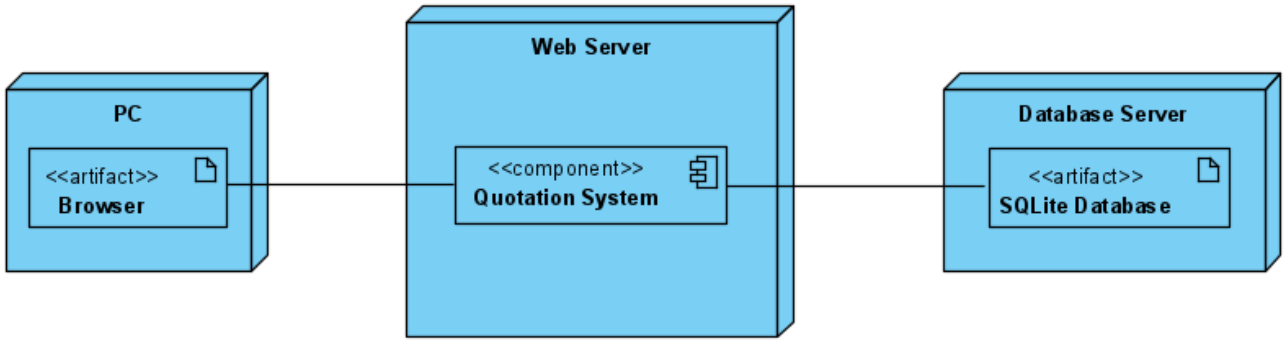


4.5 Main Components



This Component diagram shows all the functions for each Actor. This Component diagram shows the relation between each component in this quotation system.

4.6 Deployment Diagram



This Deployment Diagram shows the relationship between the user's PC Browser and our Quotation System on the Web Server. The system also establishes a connection with Django's SQLite Database to fetch and retrieve data.

5 Implementation

5.1 Development Environment

<TO DO: Describe the tools and programming elements here. Place screenshots of the IDE and solution explorer or project folder contents to illustrate the completed software.>

This project uses Visual Studio Code as the integrated development environment (IDE) and Django as the web framework. For database management, use built-in tools like Django ORM, Django's built-in object-relational mapping (ORM) system. You can use Python code to interact with your database without writing raw SQL queries.

For testing, use a testing tool such as the Django Test Framework. An integration testing framework provided by Django that allows you to create unit and integration tests for your application. You can use it to test various components of your application (views, models, forms, etc.) to make sure they are working as expected.

With these tools, you can efficiently manage your project's database and test your application to ensure that it is working correctly. The Django ORM makes it easy to interact with your database, and the Django Test Framework lets you write tests to make sure your application works as expected.

5.2 Software Integration

<TO DO: Describe the integration of the components from all group members, specify the main files updated for the integration and place source codes here.>

MANAGER

views.py

```
@login_required
def button1(request):
    """Renders the about page."""
    my_data = Sale.objects.all()
    #my_data = Sale.objects.get (quote_id=quote_id)
    return render(request, 'app/button1.html', {'my_data': my_data})

@login_required
def button2view(request):
    """Renders the about page."""
    my_data = Purchaseorderreport.objects.all()
    return render(request, 'app/button2view.html', {'my_data': my_data})

def saved(request):
    assert isinstance(request, HttpRequest)
    return render(request,'app/saved.html')



def report(request):
   if request.method =="POST":
       Fromdate=request.POST.get('Fromdate')
       Todate=request.POST.get('Todate')


searchresult=Purchaseorderreport.objects.filter(PO_date__range=(Fromdate,Tod
ate))

       return render (request, 'app/report.html',{'displaydata'
:searchresult})
   else:
       displaydata = Purchaseorderreport.objects.all()

       return render(request,'app/report.html',{'displaydata':displaydata})

def approve(request, quote_id):
    data = Sale.objects.get (quote_id=quote_id)
    item = Sale.objects.all()
```

```
    if request.method == 'POST':

        quote_id = request.POST['quote_id']
        salesstaff_id = request.POST['salesstaff_id']
        cust_id = request.POST['cust_id']
        pr_id = request.POST['pr_id']
        quot_date = request.POST['quot_date']
        quot_date = datetime.strptime(quot_date, '%b. %d,
%Y').strftime('%Y-%m-%d')
        quo_tprice = request.POST['quo_tprice']
        status = request.POST['status']

        data.quote_id=quote_id
        data.salesstaff_id=salesstaff_id
        data.cust_id=cust_id
        data.pr_id=pr_id
        data.quot_date=quot_date
        data.quo_tprice=quo_tprice
        data.status=status
        data.save()



        return redirect ('/menu')
    return render(request, 'app/approve.html', {'data': data, 'item' :item})
```

models.py

```
class Purchaseorderreport(models.Model):
    PO_id =models.IntegerField(primary_key = True, default ="")
    finance_id =models.IntegerField(default ="")
    PO_date =models.DateField()
    PO_tprice =models.IntegerField(default ="")
    def __str__(self):
        return str(self.PO_id)
```

urls.py

```
from django.contrib import admin
from django.urls import path, re_path
from app import views, forms
import django.contrib.auth.views
from django.contrib.auth.views import LoginView, LogoutView
```

```
from datetime import datetime




admin.autodiscover()

urlpatterns = [
    path('admin/', admin.site.urls),
    path('approve/<int:quote_id>', views.approve, name='approve'),
    path('update/<int:quote_id>', views.saved, name='saved'),
    re_path(r'^$', views.home, name='home'),
    re_path(r'^contact$', views.contact, name='contact'),
    re_path(r'^about$', views.about, name='about'),
    re_path(r'^login/$',
        LoginView.as_view(template_name = 'app/login.html'),
        name='login'),
    re_path(r'^logout$',
        LogoutView.as_view(template_name = 'app/index.html'),
        name='logout'),
    re_path(r'^menu$', views.menu, name='menu'),
    re_path(r'^button1$', views.button1, name='button1'),
    re_path(r'^button2view$', views.button2view, name='button2view'),
    re_path(r'saved', views.saved, name='saved'),
    re_path(r'report', views.report, name='report'),
    re_path(r'CreateQuotation',views.createQuotation,name='Create
Quotation'),
    re_path(r'ViewQuotation',views.viewQuotation,name='View Quotation'),

re_path(r'CreatePurchaseRequisition',views.CreatePurchaseRequisition,name='C
reate Purchase Requisition'),
    re_path(r'ViewPurchaseRequisition',views.viewPR,name='View Purchase
Requisition'),



]
```

Salesman

view.py

```
def createQuotation(request):
    if request.method == 'POST':
        quote_id = request.POST['quote_id']
        salesstaff_id = request.POST['salesstaff_id']
        cust_id = request.POST['cust_id']
        pr_id = request.POST['pr_id']

        qitem =Item.objects.get(item_id=request.POST['qitem'])

        quot_date = request.POST['quot_date']
        quo_tprice = request.POST['quo_tprice']
        status = request.POST['status']
              Sale1  =  Sale(quote_id=quote_id,  salesstaff_id=salesstaff_id,
cust_id=    cust_id,    pr_id=pr_id,    qitem=qitem,    quot_date=quot_date,
quo_tprice=quo_tprice,status=status)
        Sale1.save()
        return redirect('/menu')
    else:
        return render(request, 'ms_app/create_quotation.html')

def viewQuotation(request):
    sales_data = Sale.objects.all()
       return  render(request,  'ms_app/view_quotation.html',  {'sales_data':
sales_data})
```

```
def viewPR(request):
    pr_data = PurchaseRequisition.objects.all()
          return   render(request,   'ms_app/view_purchase_requisition.html',
{'pr_data': pr_data})
```

models.py

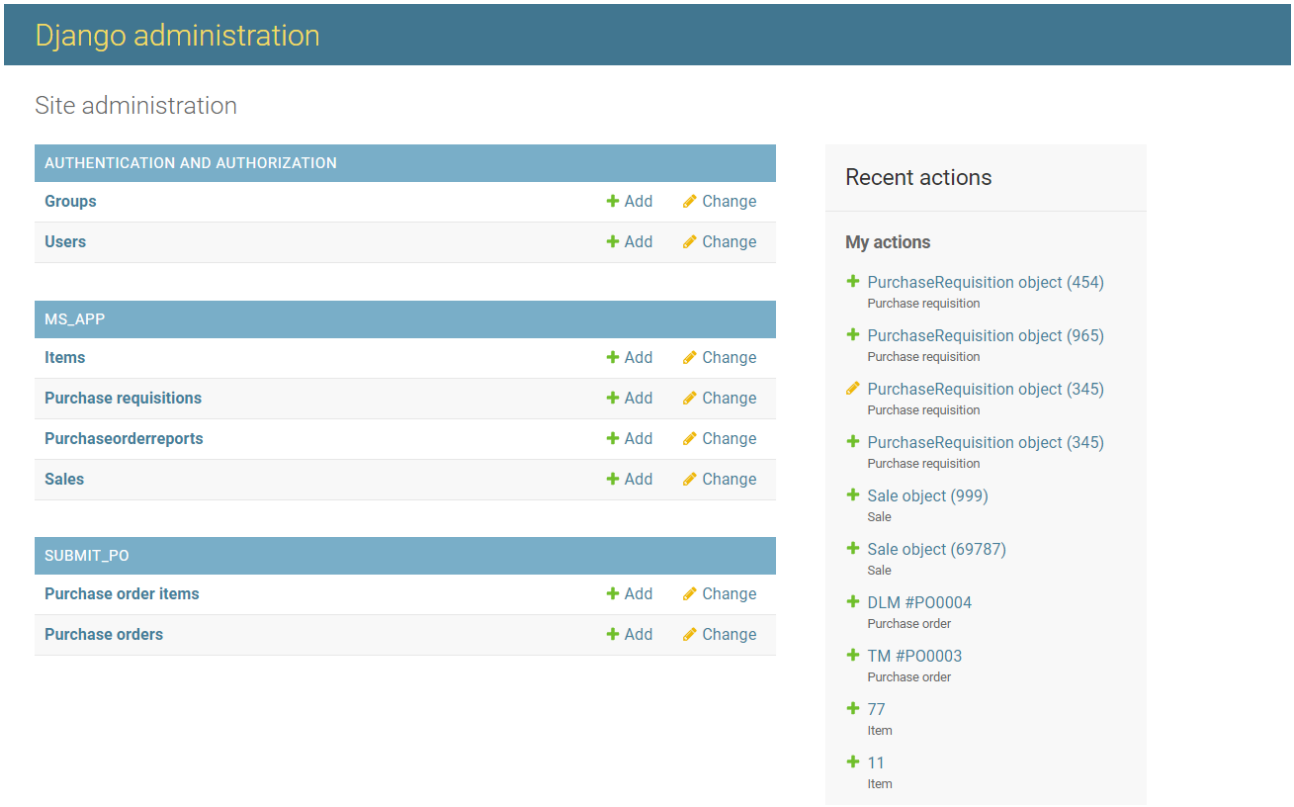
```
class Sale(models.Model):
    quote_id = models.IntegerField(primary_key=True)
    salesstaff_id = models.IntegerField()
    cust_id = models.IntegerField()
    pr_id = models.IntegerField()
```

```
      qitem  =  models.ForeignKey(Item,  on_delete=models.SET_NULL,  null=True,
blank=True)
    quot_date = models.DateField()
    quo_tprice = models.IntegerField()
    status = models.CharField(max_length=255,default='pending')

class PurchaseRequisition(models.Model):
    purchase_id = models.IntegerField(primary_key=True)
    customer_id = models.IntegerField()
      pitem  =  models.ForeignKey(Item,  on_delete=models.SET_NULL,  null=True,
blank=True)
    pr_date = models.DateField()
```

urls.py

```
 re_path(r'CreateQuotation',views.createQuotation,name='Create Quotation'),
    re_path(r'ViewQuotation',views.viewQuotation,name='View Quotation'),

re_path(r'CreatePurchaseRequisition',views.CreatePurchaseRequisition,name='C
reate Purchase Requisition'),
        re_path(r'ViewPurchaseRequisition',views.viewPR,name='View  Purchase
Requisition'),
```

5.3 Database



We used SQLite3 as the database.

6 Testing

6.1 Testing Strategy

Interrogation Testing

Interrogation Testing 1
Fawzul - Manager module
Ranjani - Finance officer module
Srihari - Salesman module

Interrogation Testing 2
Fawzul - Manager module + Finance officer module
Ranjani - Finance officer module + Salesman module
Srihari - Salesman module + Manager module

Interrogation Testing 2
Fawzul - Manager module + Finance officer module + Salesman module
Ranjani - Finance officer module + Salesman module + Manager module
Srihari - Salesman module + Manager module + Finance officer module

Unit Testing

Fawzul - Manager module
- *View Quotation*
- *Approve Quotation*
- *View Purchase Order Report*

Srihari - Salesman module
- *Create quotation*
- *view quotation*
- *view purchase requisition*

Ranjani - Finance module
- *Submit Purchase Order*
- *View Purchase Order*

6.2 Test Data

6.2.1 Test Data Set 1

Quotation

QUOTE ID	***SALESSTAFF ID***	***CUST ID***	***PR ID***	***QITEM***	***QUOT DATE***	***QUO TPRICE***	***STATUS***
69787	*3433*	*2425*	*55677*	*77*	*Jan. 9, 2023*	*847*	*pending*
999	*453*	*242*	244	11	*Jan.10, 2023*	*454*	*rejected*
799	*54546*	*3535*	*35435*	11	*Jan. 2, 2023*	*43535*	*approve*

Purchase order/ Report

Officer:	Date	**Company name:**	**Address line 1:**	**Address line 2:**	**City:**	**State:**	**Zipcode:**	**Contact name:**	**Contact no:**
john-doe	Jan. 26, 2023, 8:15 p.m.	Planergy	no45, Taman Megah	jalan 2/3	Kepong	Wilayah Persekutuan	53990	Rachea l	039888689
john-doe	Jan. 27, 2023, 8:21 a.m.	TM	No 176, Jalan Ampang	Kuala Lumpur	Wilayah Persekutuan	Wilayah Persekutuan	57100	Edward	0137676864
john-doe	Jan. 27, 2023, 8:22 a.m.	DLM	Boston Office	One Post Office Square, Suite 3600	Boston MA	USA	02109	Patrick	800-145-8259

Purchase Requisition

Purchase Requisition ID	Customer ID	Item ID	Date
345	567	11	2023-01-09
965	111	77	2023-01-11
454	657	11	2023-01-02

6.2.2 Test Data Set 2

Quotation

QUOTE ID	***SALESSTAFF ID***	***CUST ID***	***PR ID***	***QITEM***	***QUOT DATE***	***QUO TPRICE***	***STATUS***
23	*12*	*24*	*45*	*23*	*JJan.1, 2023*	*200*	*pending*
89	*18*	*34*	312	45	*Jan.12, 2023*	*300*	*pending*
27	*17*	*87*	*412*	34	*Jan. 9, 2023*	*400*	*pending*

Purchase order/ Report

Officer:	Date	**Company name:**	**Address line 1:**	**Address line 2:**	**City:**	**State:**	**Zipcode:**	**Contact name:**	**Contact no:**
john-doe	Jan. 28, 2023, 8:15 p.m.	HP	no45, Taman Gombak	Batu Canes	Kepong	Selangor	68100	Salman	0361790866
john-doe	Jan. 29, 2023, 8:21 a.m.	DELL	No 16, Jalan Selayang	Batu Caves	Wilayah Persekutuan	Selangor	68100	Rahman	0122133456
john-doe	Jan. 13, 2023, 8:22 a.m.	LENOVO	No 6 Taman selaseh	Selaseh	Greenwood	Selangor	68100	Rahim	0192233242

Purchase Requisition

Purchase Requisition ID	Customer ID	Item ID	Date
214	333	23	2023-01-10
212	444	45	2023-01-11
241	222	34	2023-01-12

6.2.3 Test Data Set 3

Finance Officer

username	password	first_name	last_name	group	email
john-doe	jdjdjdjd	John	Doe	Finance Officer	john@quosys.com

Purchase Order

Officer:	Company name:	Address line 1:	Address line 2:	City:	State:	Country	Zipcode:	Contact name:	Contact no:
john-doe	DLM	No, 45, Taman Megah	Jalan 2/3	Kepong	Selangor	Malaysia	53990	Racheal	03-988-8689
john-doe	TM	No 176, Taman Tekno	Jalan Ampang	Wilayah Persekutuan	Selangor	Malaysia	57100	Edward	013-767-6864
john-doe	Plan ergy	Boston	One Post	Boston	MA	USA	02109	Patrick	800-145-8259

		Office	Office Square, Suite 3600						

Purchase Order Item

Name	Price	Quantity
Economy Manilla Envelopes - 500	15.49	2
Viking A4 Economy Copier	3.59	5
3 Tier Letter Tray	23.89	3
Tettley Tea Round Tea Bags 440/pk	20.49	1
Nescafe Gold Blend Coffee 7oz	34.99	1

6.3 Acceptance Tests

6.3.1 Manager Acceptance Test

Criteria	Fulfilled?	Remarks
The manager must be able to approve the correct quotation.	Yes	
The system must be able to display all the quotations submitted by the salesman.	Yes	
The system must be able to display all the purchase order details	Partial	Should have more details of P.O.s e.g. summary
The system should categorize purchase orders correctly according to a specific period selected by the manager..	Yes	

The system should allow the manager to login.	Yes	

Date tested :28/1/2023

% Complete : 90%

Tested by : Ranjani

Verified by : Dr. Yeoh

6.3.2 Salesman Acceptance Test

Criteria	**Fulfilled?**	**Remarks**
Create Quotation and save in the database	Partial	Only 1 item, no item quantity
View Quotation from the database	Yes	
View Purchase Requisition from the database	Yes	

Date tested : 28/1/23
% Complete : 80%
Tested by : Fawzul Bari
Verified by : Dr. Yeoh

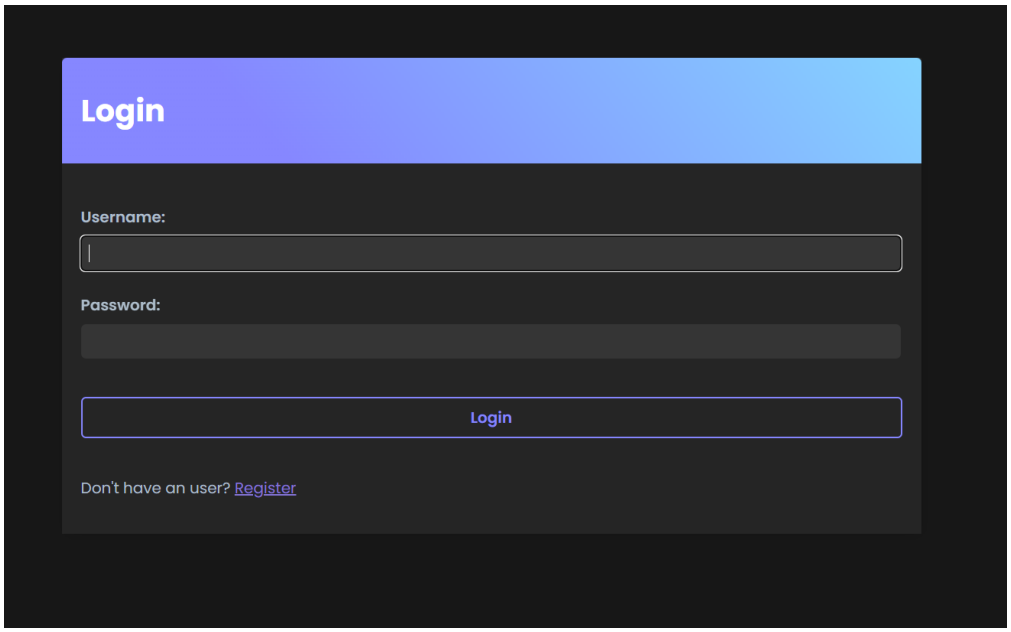
6.3.3 Finance Officer Acceptance Test

Criteria	Fulfilled?	Remarks
The system should display the Finance Officer's Purchase Order Dashboard upon login.	Yes	
The system should display all key-ins/submissions of Purchase Order by the Finance Officer.	Yes	
The system should display the option for the Finance Officer to key-in a new Purchase Order to the system.	Yes	
The system should allow the Finance Officer to key-in the details of a Purchase Order and submit it.	Yes	

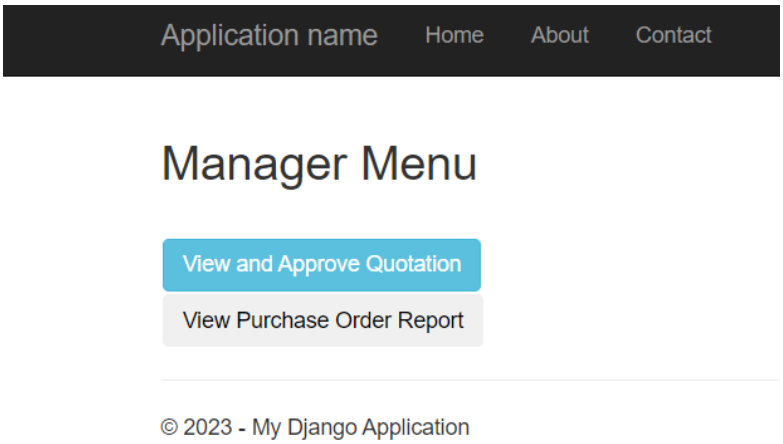
Date tested : 28/1/23
% Complete : 100%
Tested by : Srihari Naidu
Verified by : Dr. Yeoh

7 Sample Screens

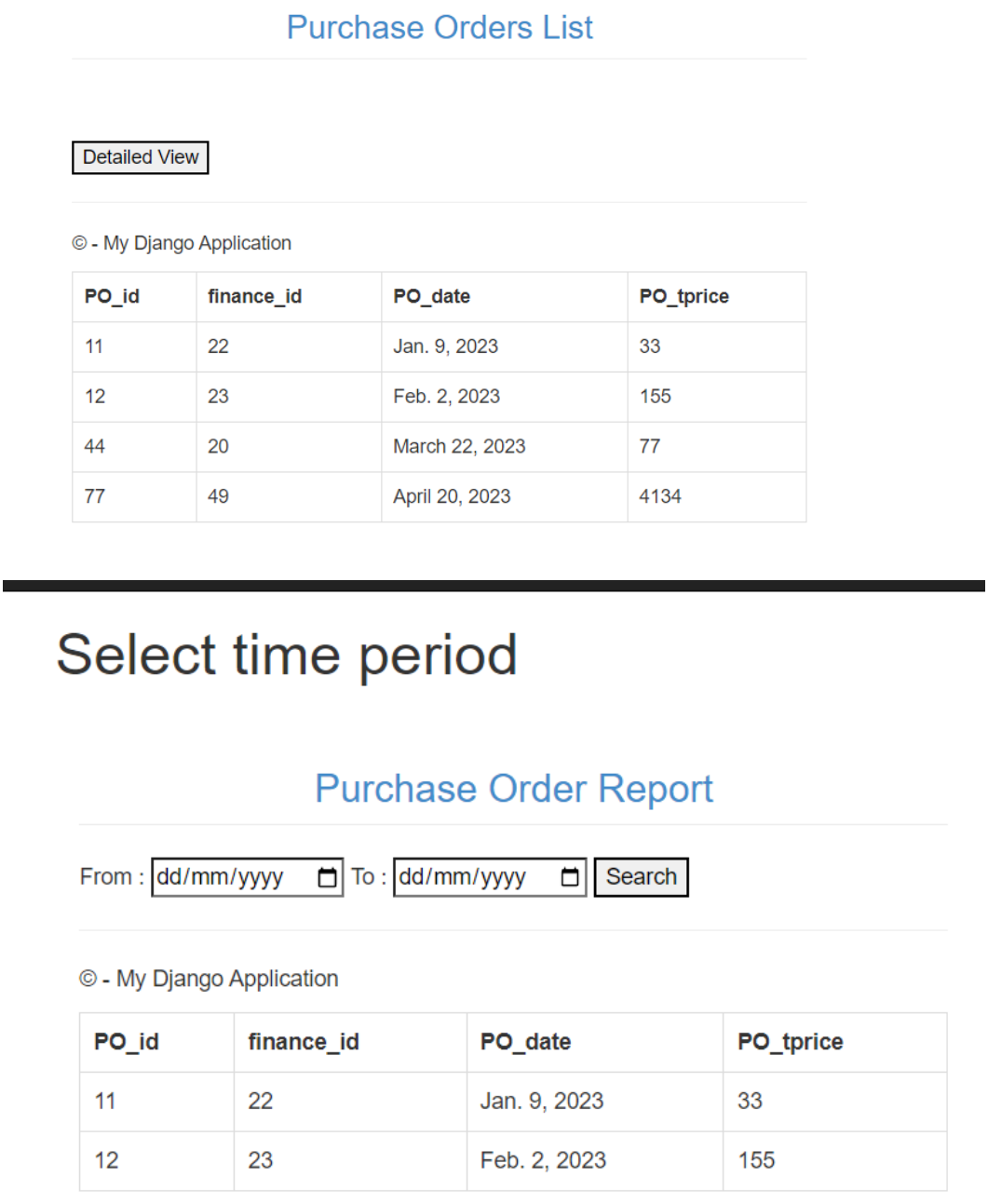
7.1 Main Screens



Manager Home Screen



Purchase Order Report view screen



Quotation List

© - My Django Application

quote_id	salesstaff_id	cust_id	pr_id	qitem_id	quot_date	quo_tprice	status	Update
4	5	6	3	23	Jan. 17, 2023	67	pending	approve
7	4	2	5	88	Jan. 9, 2023	33	pending	approve

quote_id

345

salesstaff_id

3445

cust_id

3454

pr_id

3345

qitem

11

quot_date

Jan. 13, 2023

quo_tprice

6767

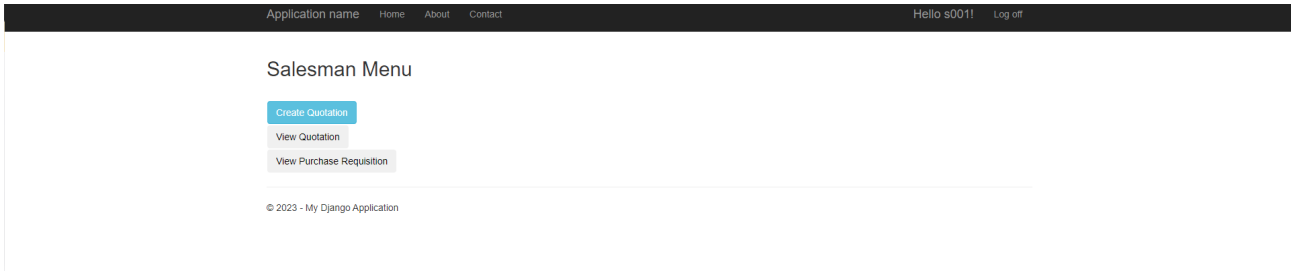
status

approved

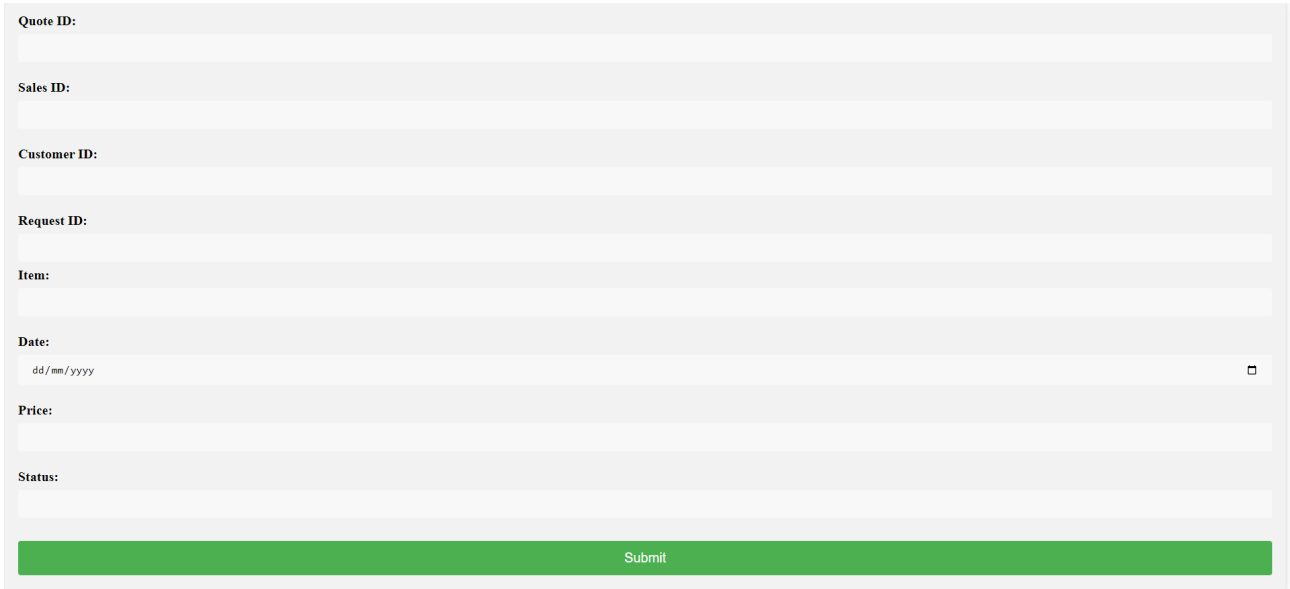
Update

7.2 Salesman

Salesman Home Screen



Create Quotation Screen



View Quotation Screen

Quotation Table

Quote ID	Sales ID	Customer ID	Request ID	Item	Date	Price	Status
4444	12345	1111	dell	2222	Jan. 18, 2023	1000	pending
4444	1111	2222	1111	22222	Jan. 18, 2023	222	pending
4444	3444	134521	34555	dell	Jan. 14, 2023	333	pending
1111	2222	3333	4444	Tablet	Jan. 19, 2023	300	pending
1211	2233	1123	11233	13331	Jan. 18, 2023	23411	sent

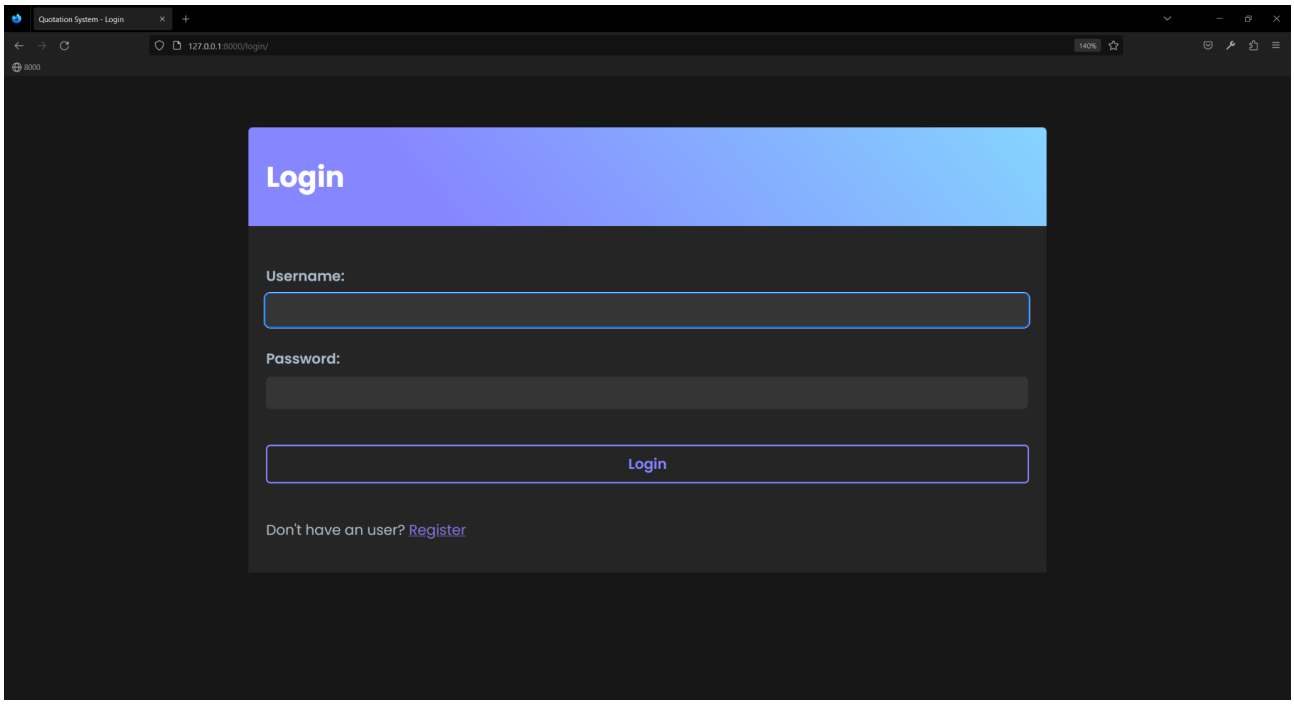
View Purchase Requisition Screen

View Purchase Requisition

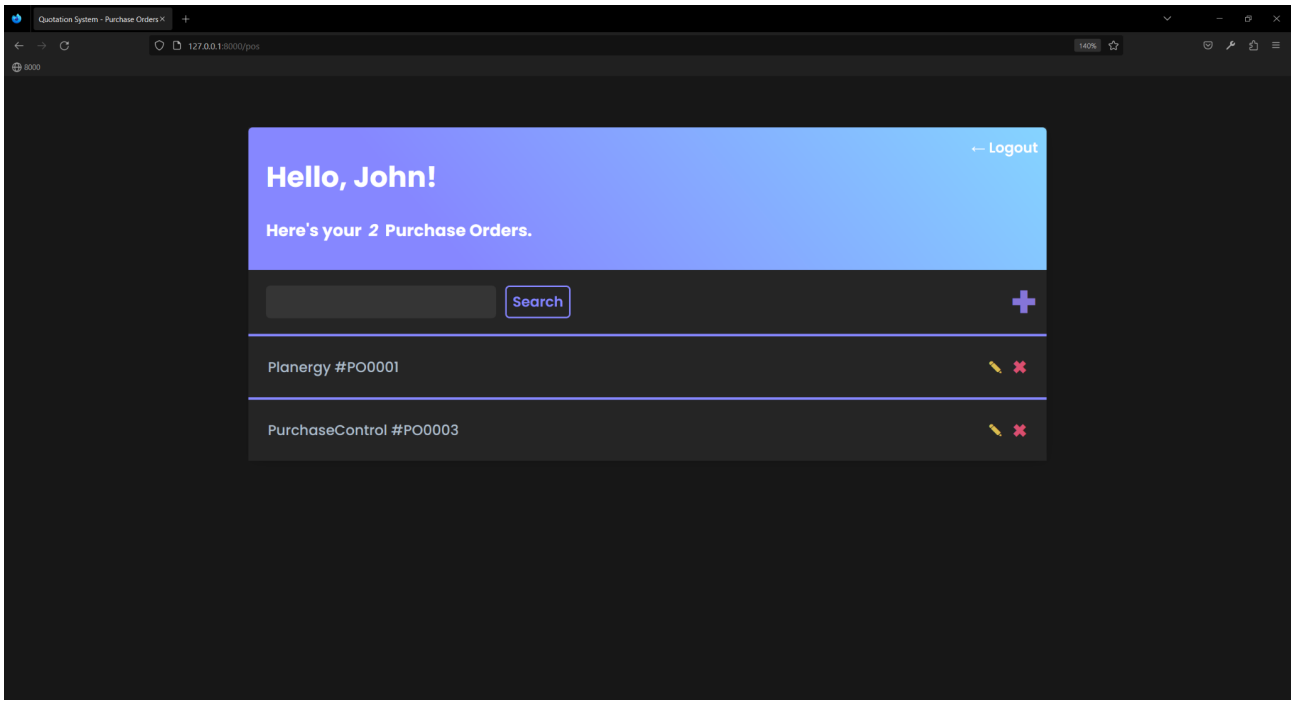
PR ID	Customer ID	PR Item ID	PR Date
23	45	77	Jan. 2, 2023
45	23	11	Jan. 4, 2023

7.3 Finance Officer

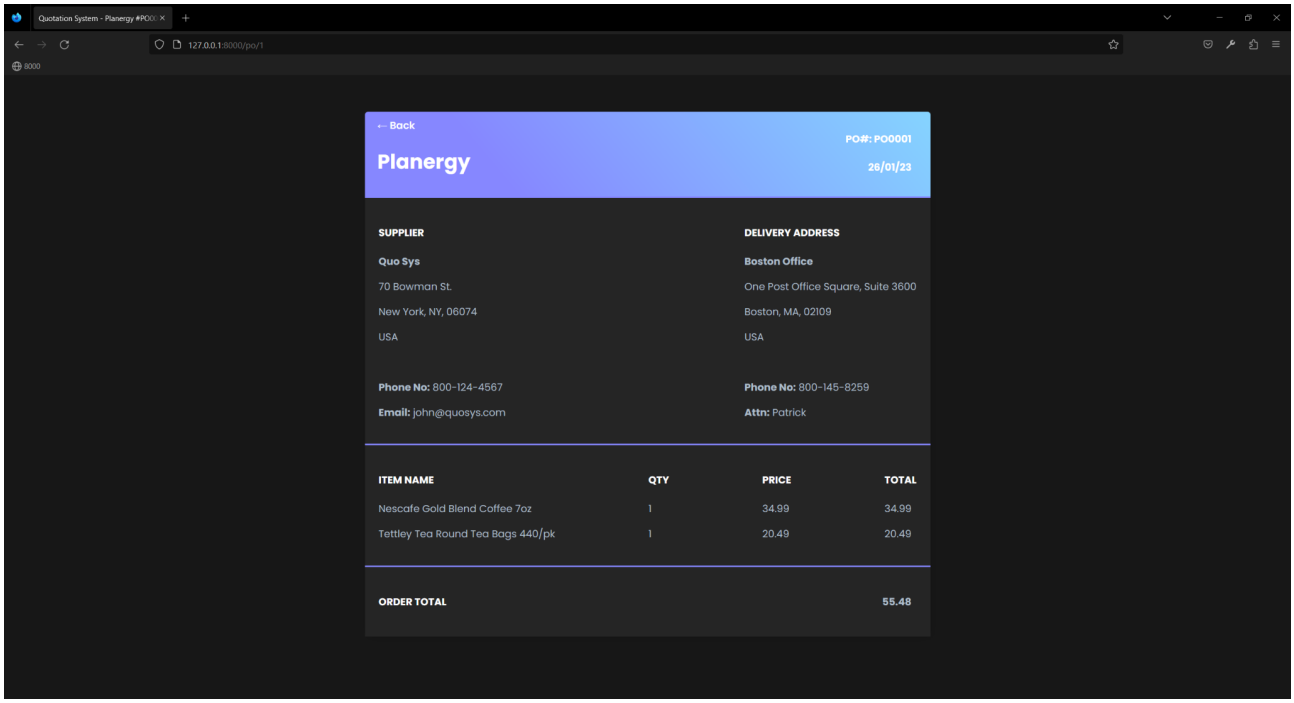
1.1.1 Login



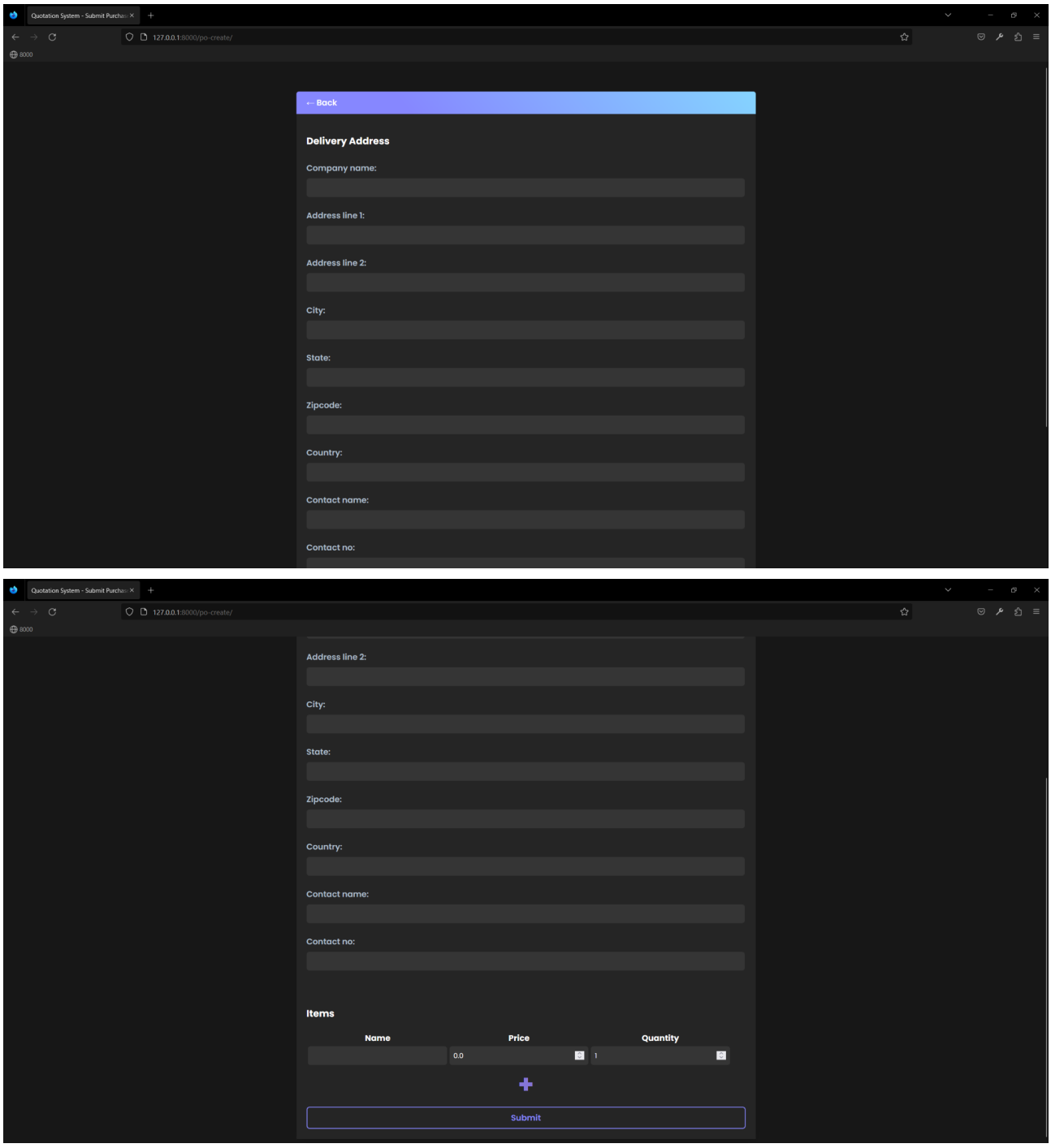
1.1.2 View Purchase Order List



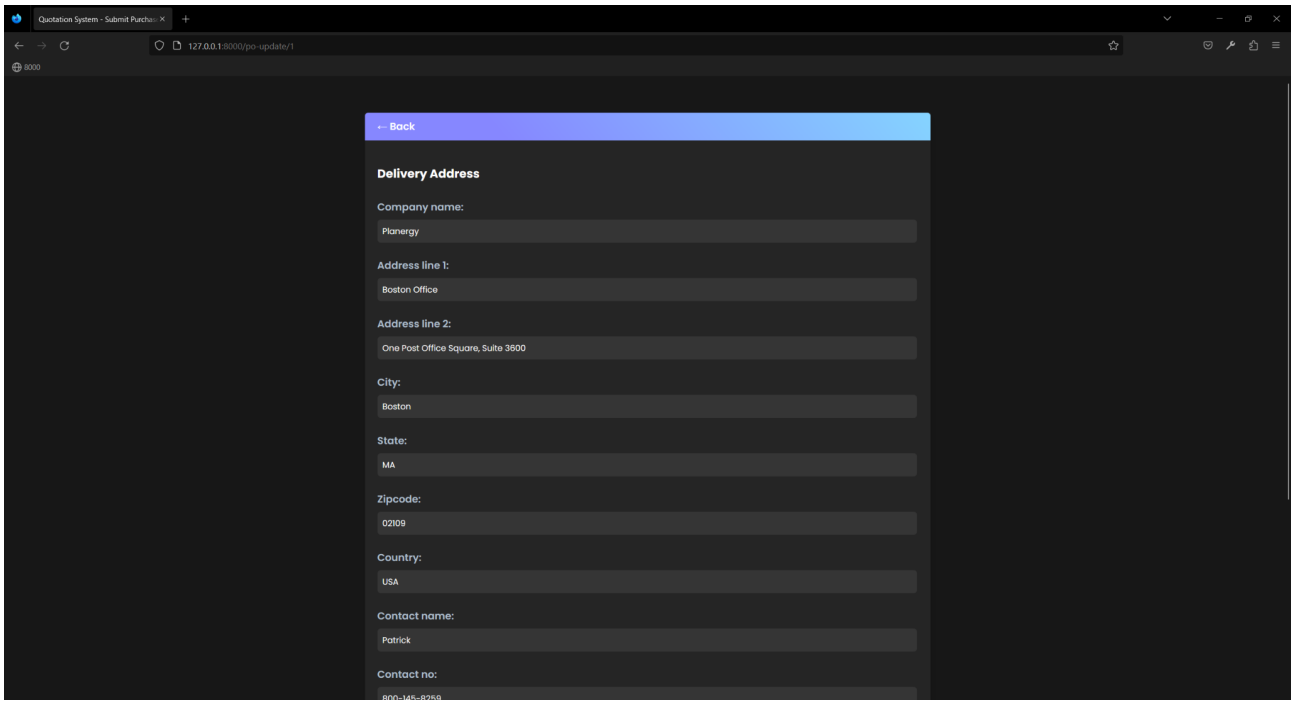
1.1.3 View Purchase Order

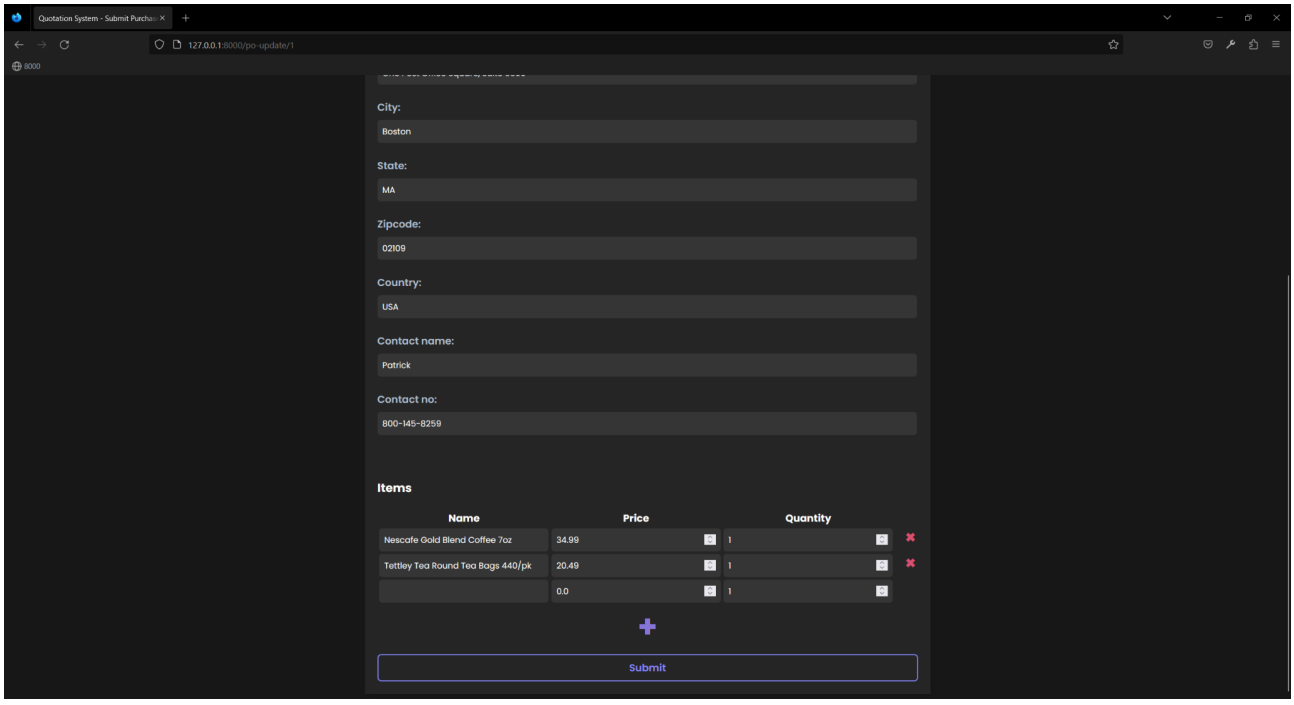


1.1.4 Create Purchase Order

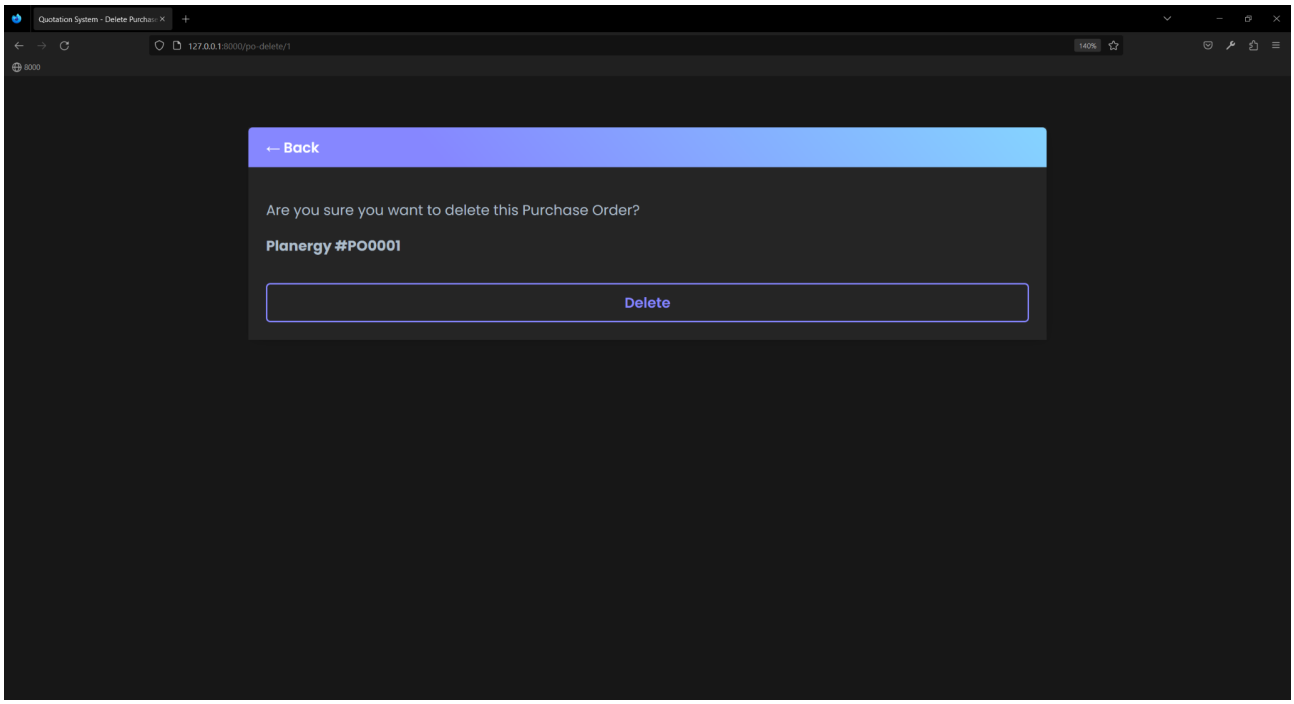


1.1.5 Edit Purchase Order





1.1.6 Delete Purchase Order



8 Conclusion

8.1 Completion of Software

The following stages are necessary to complete software that has been developed with three actors, a manager, a salesman, and a finance officer:

Creating the necessary classes, functions, and modules to manage the functionality required by each actor is part of the design and development of the software. The software would be made to store information about users, transactions, and other pertinent data in the database.

Database design and data modelling In this step, the essential tables for storing the data are created as well as the database structure. The database would be created to record the information for each actor, including the salesperson, manager, and finance officer.

8.2 Software Quality Assurance

We tried to handle the exception well when there is any input mismatch and we ensured that all the urls are working. we Developed and maintained a comprehensive test plan and conducted unit testing. We also Conducted inspections and walkthroughs of software productsMaintainability is one of the crucial quality factors I emphasise in my software development projects. I think that the long-term viability of a project depends on the ability to maintain and upgrade software with ease. I adhere to a variety of best practices, such as adopting clear and consistent coding rules, to guarantee maintainability.

8.3 Group Collaboration

We made care to hold frequent team meetings to keep in sync about progress and any roadblocks, and we made sure that each team member had clear roles and duties from the start. Additionally, we had to meet a lot of deadlines while remaining adaptable and sensitive to changes in the demands. We employed agile approaches to address issue and were successful in delivering the software on schedule. I think that one of the reasons for our success was the teamwork and trust we developed.

8.4 Problems Encountered

It was challenging for us to comprehend the template and to code because we had no prior knowledge with the python programming language or with HTML. We were able to develop this method after using numerous sources and YouTube lessons. Another problem that prevented us from finishing this job was a lack of time. to continue development. We had to spend a lot of time gathering data and attempting to comprehend the requirements and goals for the product. Due to these delays in the project timetable, development took longer than expected.

8.5 Remarks/Comments

It was a good learning experience and we learned new things.
However, there could have also been a few difficulties or problems that emerged throughout development, just as with every project. Django applications frequently run into problems with migration of databases and configuring the app for use in a production environment.

9 User Guide

*<This section is **Optional**. Write a user guide on how to use the system and place it here.>*