



Sri Lanka Institute of Information Technology

SE3030 - Software Architecture [2025/FEB]

ASSIGNMENT 01

OSGI FRAMEWORK FOR PAINT SHOP

Submitted by

Name	REGISTRATION NUMBER	Status	Batch
Senarathne S M A D	IT20089436	Pro-rata	Weekend

Git Hub Repository Link : https://github.com/amila1998/OSGI_FrameWork-Paint-Shop

Contents

OSGI FRAMEWORK FOR PAINT SHOP	1
INTRODUCTION	5
RELATIONSHIP DIAGRAM.....	6
WORKFLOWS.....	7
CUSTOMER WORKFLOW	7
ADMIN WORKFLOW	8
DRIVER WORKFLOW	9
MANIFEST IMPLEMENTATION (EXPORTED SERVICES / IMPORTED SERVICES)	10
ADMIN PRODUCER	10
MIXER PRODUCER.....	10
ORDER PRODUCER.....	11
DELIVERY PRODUCER.....	11
CUSTOMER PRODUCER.....	12
ADMIN CONSUMER	12
CUSTOMER CONSUMER.....	13
DELIVERY CONSUMER.....	13
ALL COMMANDS OF BUNDLE INSTALL AND RUN IN OSGI FRAMEWORK.....	14
Running the OSGi Bundles Without Exporting JARs (Directly from Eclipse)	14
Running the Exported OSGi JAR Bundles	20

Figures

Figure 1: OSGI Architecture	4
Figure 2: OSGI Behaviors	4
Figure 3: System Publisher and Subscriber Diagram	5
Figure 4: System Publisher and Subscriber Diagram - Relations.....	6
Figure 5: customer workflow	7
Figure 6: Admin Workflow	8
Figure 7: Driver Workflow.....	9
Figure 8: Admin producer manifest.....	10
Figure 9: Mixer producer manifest.....	10
Figure 10: Order producer manifest.....	11
Figure 11: Delivery producer manifest	11
Figure 12: Customer producer manifest.....	12
Figure 13: Admin consumer manifest	12
Figure 14: Customer consumer manifest	13
Figure 15: Delivery consumer manifest.....	13
Figure 16: Run Configurations.....	14
Figure 17: Verify Bundles.....	15
Figure 18: Admin Workflow Output SS.....	16
Figure 19: Admin Workflow Output SS.....	16
Figure 20: Admin Workflow Output SS.....	17
Figure 21: Admin Workflow Output SS.....	17
Figure 22: Customer Workflow Output SS.....	18
Figure 23: Customer Workflow Output SS.....	18
Figure 24: Customer Workflow Output SS.....	19
Figure 25: Driver Workflow Output SS	19

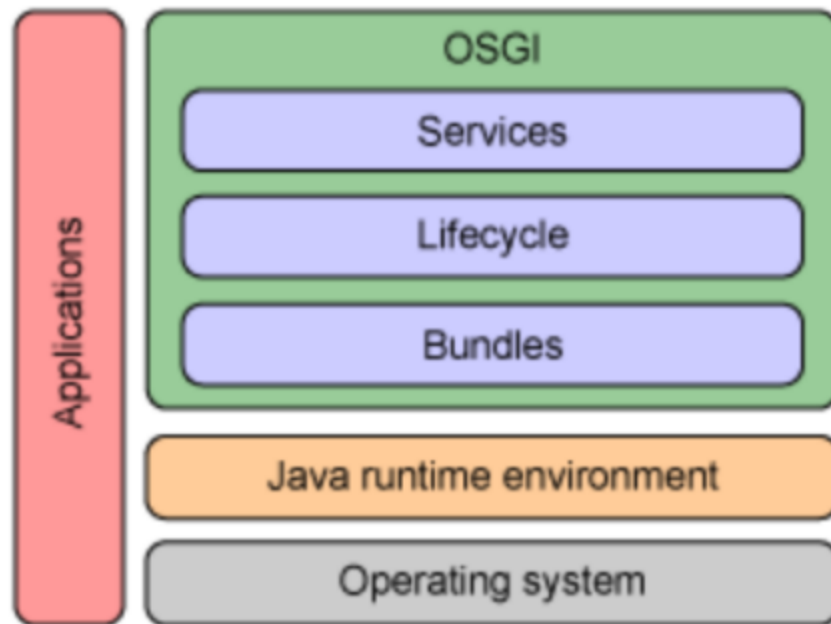


Figure 1: OSGI Architecture

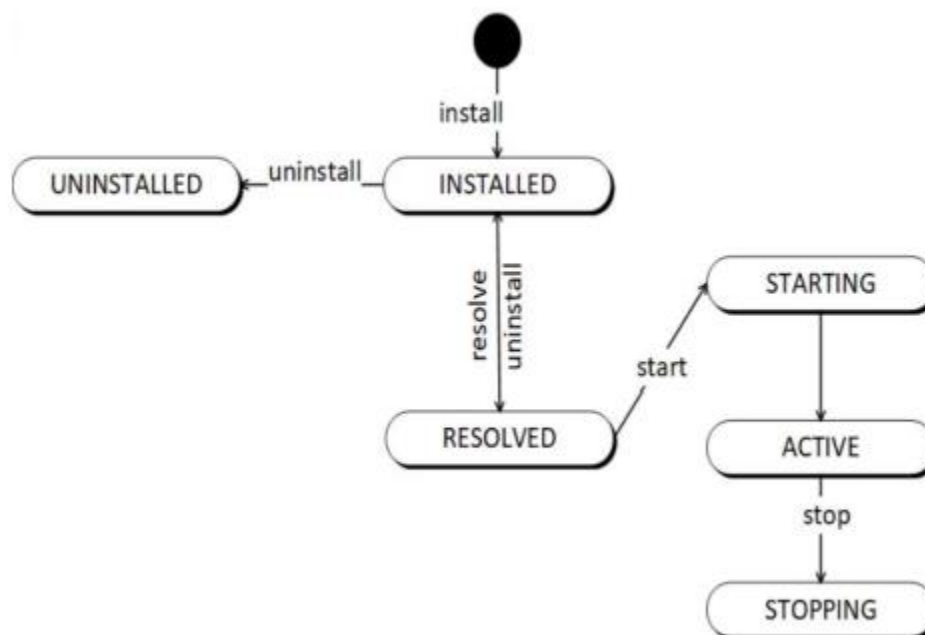


Figure 2: OSGI Behaviors

INTRODUCTION

This system is designed to efficiently manage a paint shop. Customers can sign up and log in to browse and place orders for paints. Additionally, they have the option to choose whether their orders should be delivered.

The system also includes administrative functionality, allowing administrators to create new paints and manage delivery user accounts. Delivery users can log in to view a list of pending delivery orders and process them accordingly.

Leveraging the power of the OSGi (Open Service Gateway Initiative) framework, this system provides a modular and dynamic architecture, ensuring efficient management and seamless delivery of services.

- Admin Producer
- Customer Producer
- Delivery Producer
- Mixer (Paint) Producer
- Order Producer

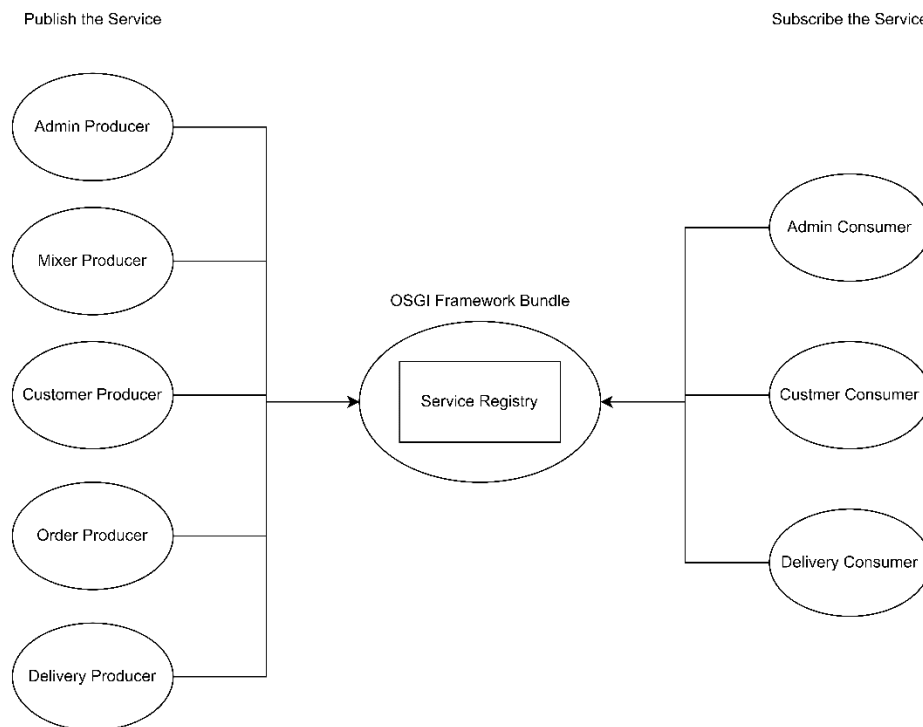


Figure 3: System Publisher and Subscriber Diagram

RELATIONSHIP DIAGRAM

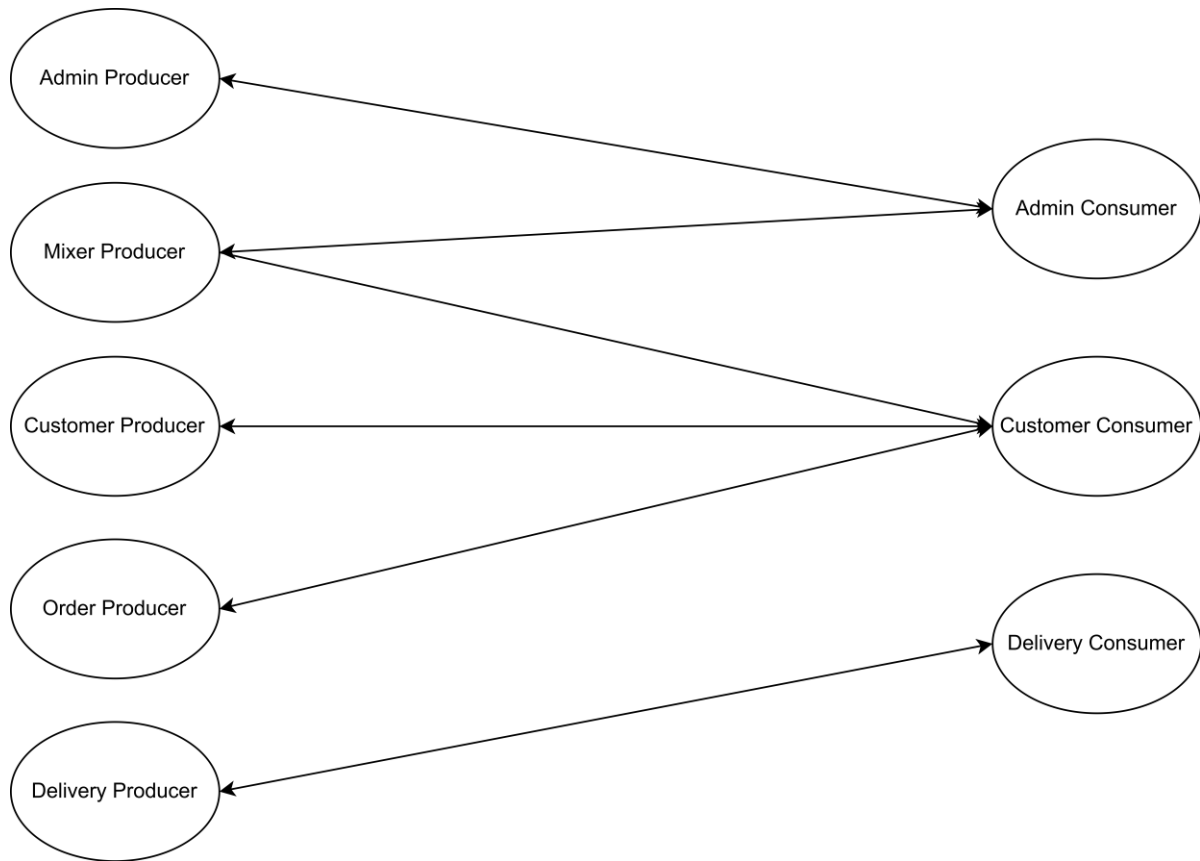


Figure 4: System Publisher and Subscriber Diagram - Relations

WORKFLOWS

CUSTOMER WORKFLOW

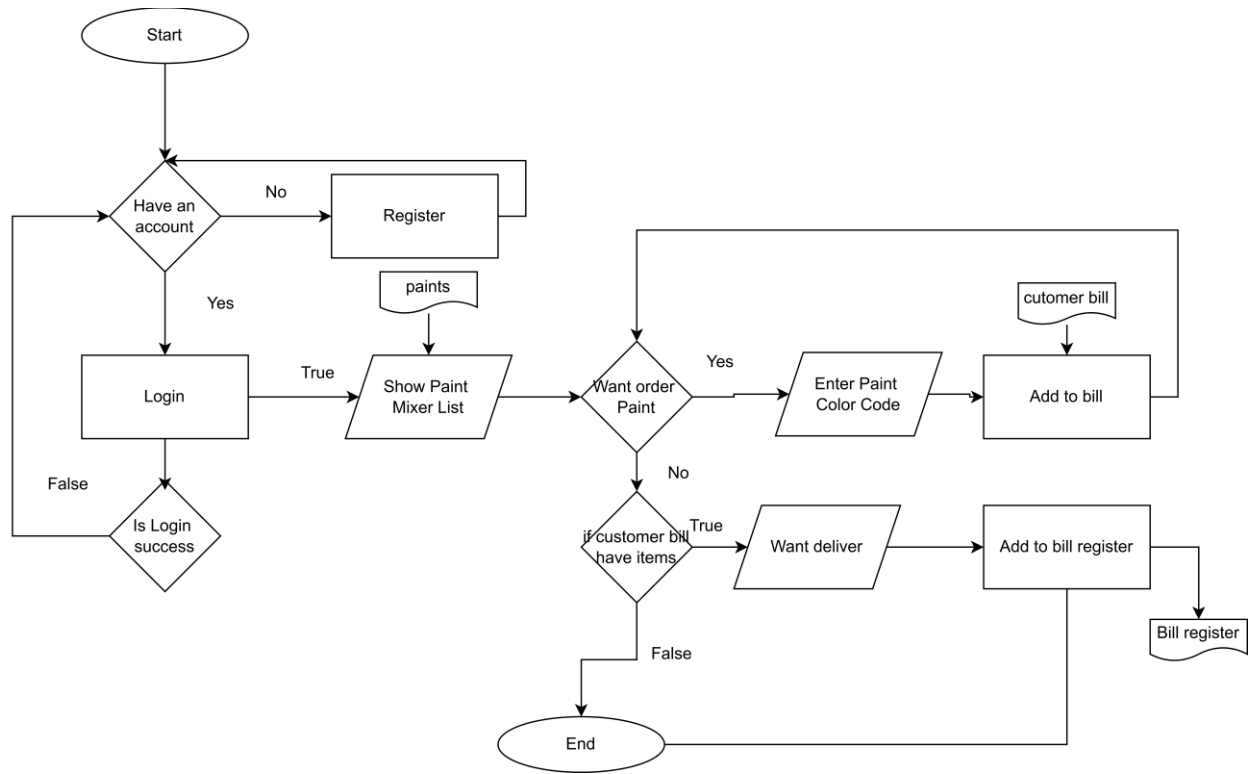


Figure 5: customer workflow

ADMIN WORKFLOW

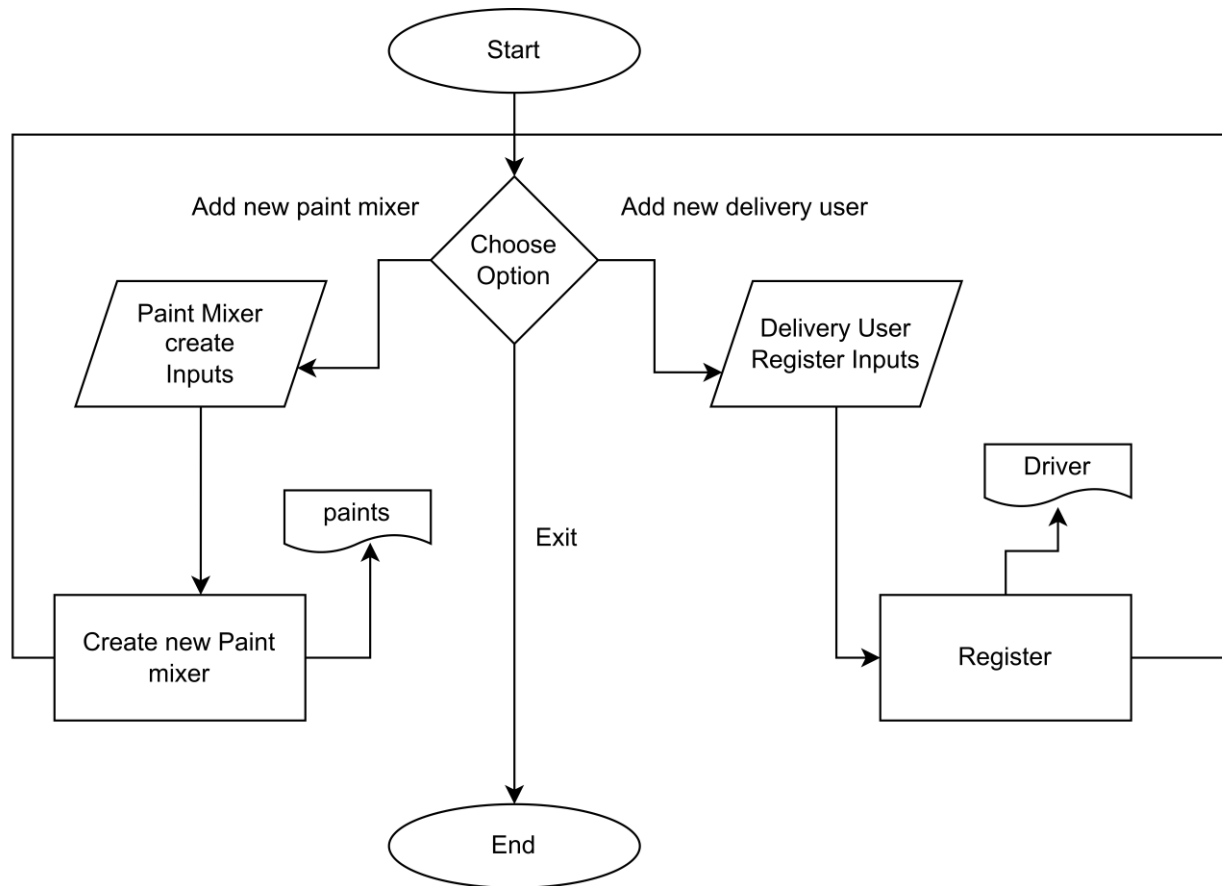


Figure 6: Admin Workflow

DRIVER WORKFLOW

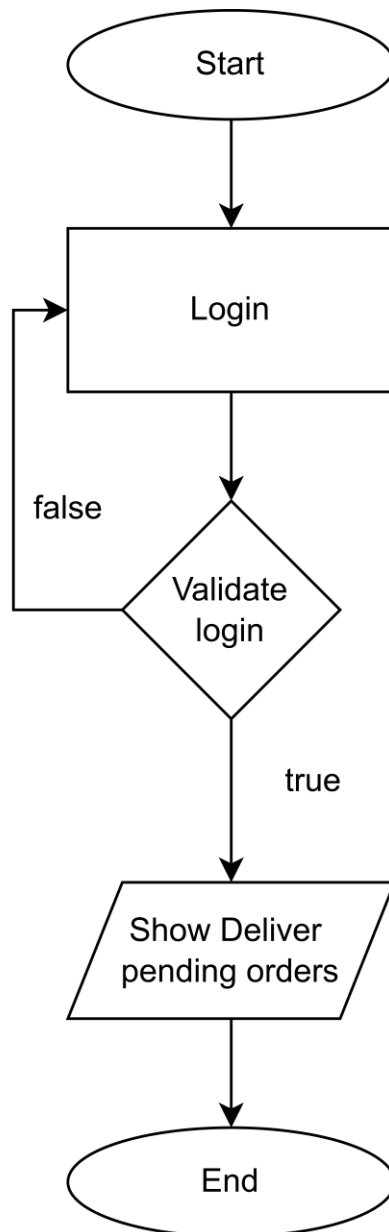
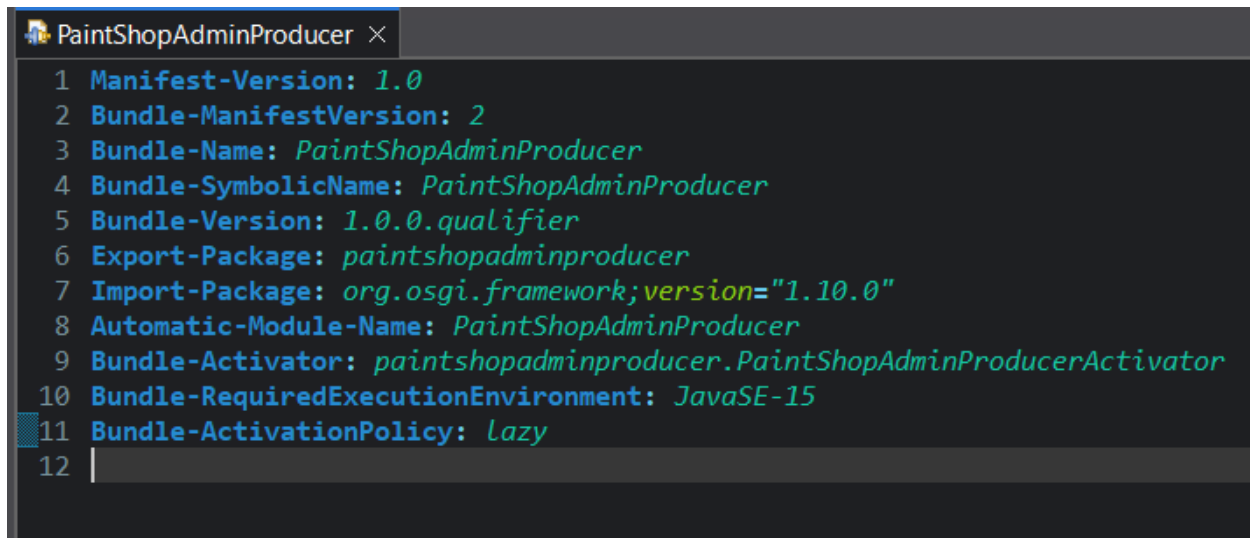


Figure 7: Driver Workflow

MANIFEST IMPLEMENTATION (EXPORTED SERVICES / IMPORTED SERVICES)

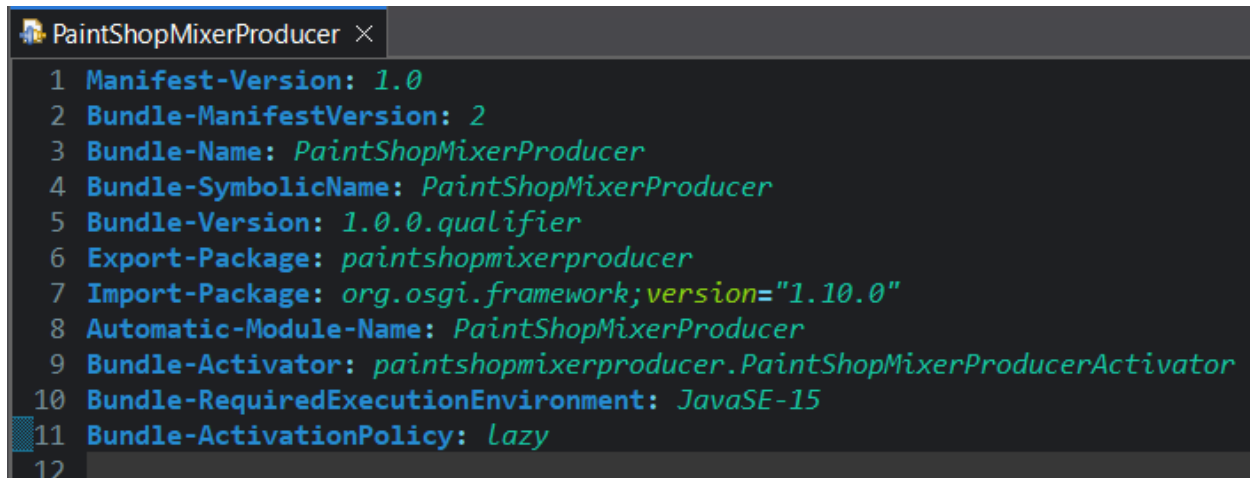
ADMIN PRODUCER



```
PaintShopAdminProducer x
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: PaintShopAdminProducer
4 Bundle-SymbolicName: PaintShopAdminProducer
5 Bundle-Version: 1.0.0.qualifier
6 Export-Package: paintshopadminproducer
7 Import-Package: org.osgi.framework;version="1.10.0"
8 Automatic-Module-Name: PaintShopAdminProducer
9 Bundle-Activator: paintshopadminproducer.PaintShopAdminProducerActivator
10 Bundle-RequiredExecutionEnvironment: JavaSE-15
11 Bundle-ActivationPolicy: lazy
12
```

Figure 8: Admin producer manifest

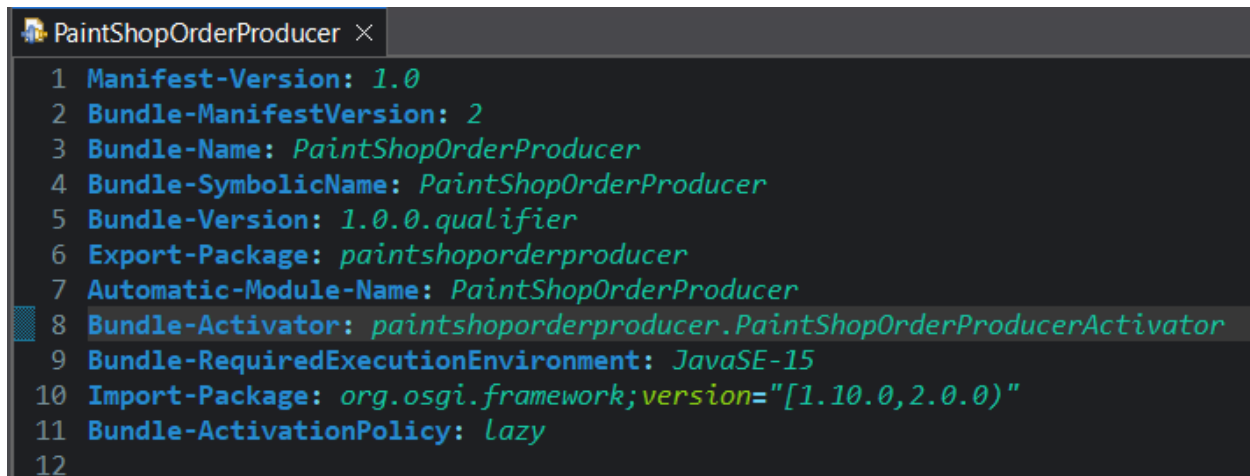
MIXER PRODUCER



```
PaintShopMixerProducer x
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: PaintShopMixerProducer
4 Bundle-SymbolicName: PaintShopMixerProducer
5 Bundle-Version: 1.0.0.qualifier
6 Export-Package: paintshopmixerproducer
7 Import-Package: org.osgi.framework;version="1.10.0"
8 Automatic-Module-Name: PaintShopMixerProducer
9 Bundle-Activator: paintshopmixerproducer.PaintShopMixerProducerActivator
10 Bundle-RequiredExecutionEnvironment: JavaSE-15
11 Bundle-ActivationPolicy: lazy
12
```

Figure 9: Mixer producer manifest

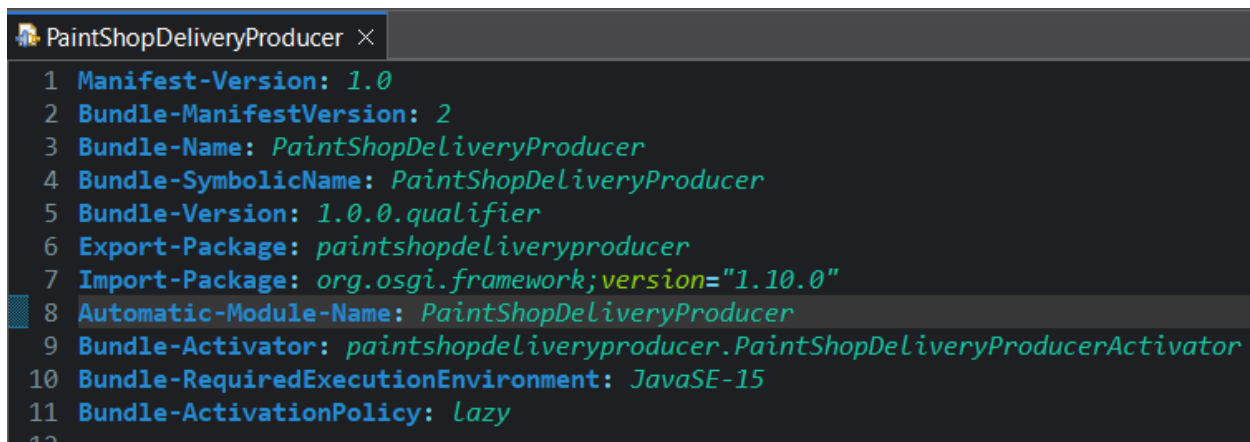
ORDER PRODUCER



```
PaintShopOrderProducer ×
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: PaintShopOrderProducer
4 Bundle-SymbolicName: PaintShopOrderProducer
5 Bundle-Version: 1.0.0.qualifier
6 Export-Package: paintshoporderproducer
7 Automatic-Module-Name: PaintShopOrderProducer
8 Bundle-Activator: paintshoporderproducer.PaintShopOrderProducerActivator
9 Bundle-RequiredExecutionEnvironment: JavaSE-15
10 Import-Package: org.osgi.framework;version="[1.10.0,2.0.0)"
11 Bundle-ActivationPolicy: lazy
12
```

Figure 10: Order producer manifest

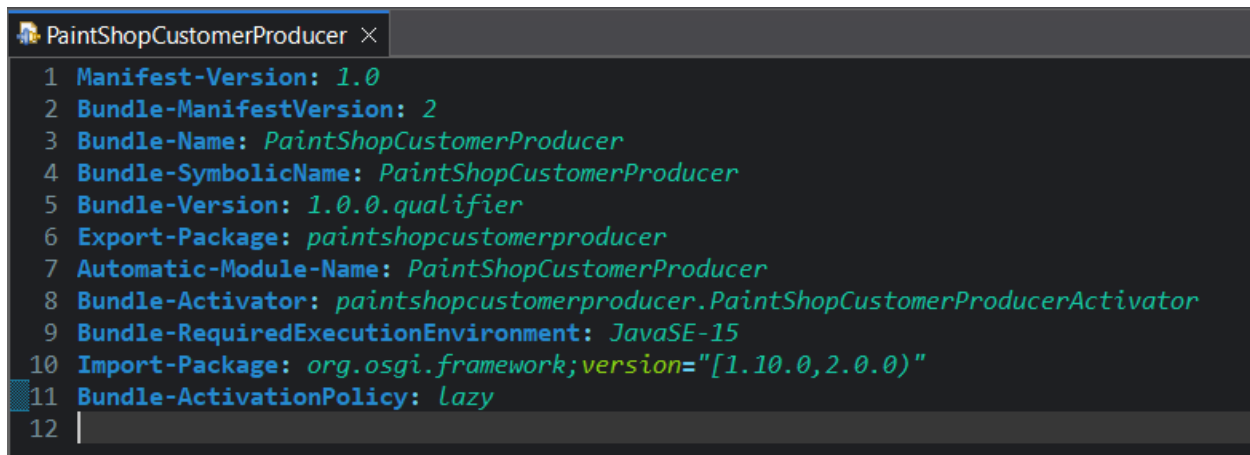
DELIVERY PRODUCER



```
PaintShopDeliveryProducer ×
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: PaintShopDeliveryProducer
4 Bundle-SymbolicName: PaintShopDeliveryProducer
5 Bundle-Version: 1.0.0.qualifier
6 Export-Package: paintshopdeliveryproducer
7 Import-Package: org.osgi.framework;version="1.10.0"
8 Automatic-Module-Name: PaintShopDeliveryProducer
9 Bundle-Activator: paintshopdeliveryproducer.PaintShopDeliveryProducerActivator
10 Bundle-RequiredExecutionEnvironment: JavaSE-15
11 Bundle-ActivationPolicy: lazy
12
```

Figure 11: Delivery producer manifest

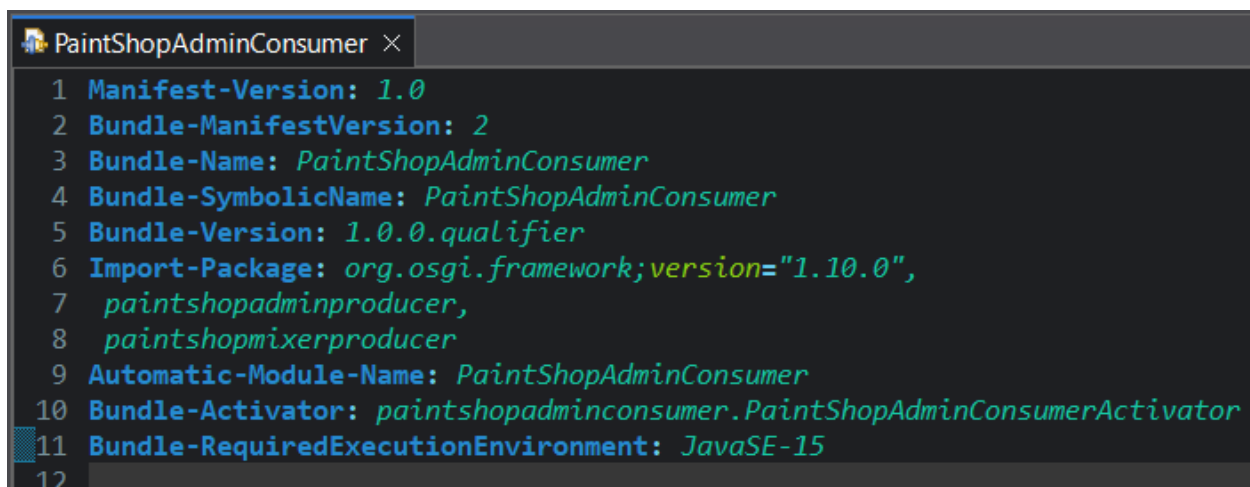
CUSTOMER PRODUCER



```
PaintShopCustomerProducer ×
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: PaintShopCustomerProducer
4 Bundle-SymbolicName: PaintShopCustomerProducer
5 Bundle-Version: 1.0.0.qualifier
6 Export-Package: paintshopcustomerproducer
7 Automatic-Module-Name: PaintShopCustomerProducer
8 Bundle-Activator: paintshopcustomerproducer.PaintShopCustomerProducerActivator
9 Bundle-RequiredExecutionEnvironment: JavaSE-15
10 Import-Package: org.osgi.framework;version="[1.10.0,2.0.0)"
11 Bundle-ActivationPolicy: lazy
12
```

Figure 12: Customer producer manifest

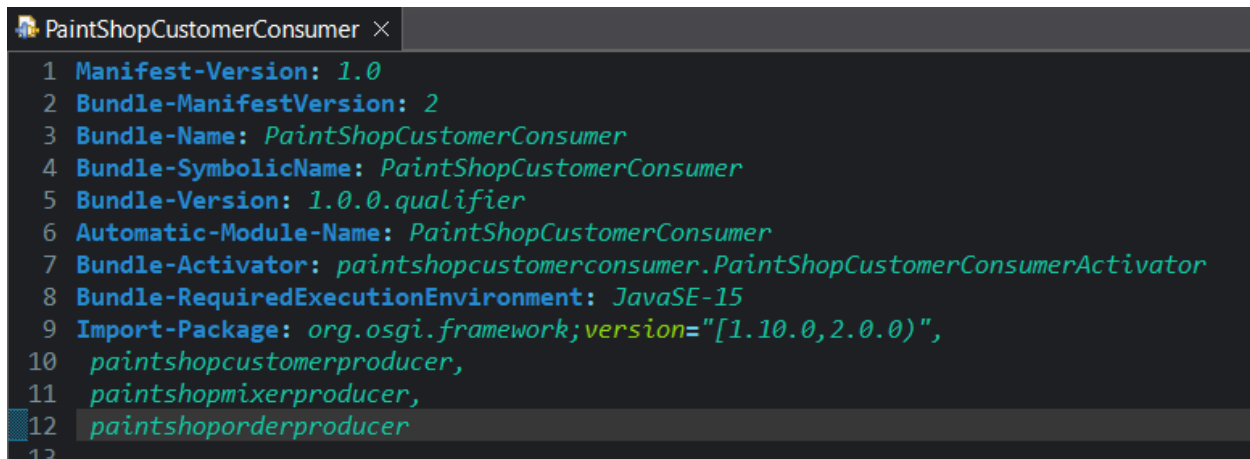
ADMIN CONSUMER



```
PaintShopAdminConsumer ×
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: PaintShopAdminConsumer
4 Bundle-SymbolicName: PaintShopAdminConsumer
5 Bundle-Version: 1.0.0.qualifier
6 Import-Package: org.osgi.framework;version="1.10.0",
7   paintshopadminproducer,
8   paintshopmixerproducer
9 Automatic-Module-Name: PaintShopAdminConsumer
10 Bundle-Activator: paintshopadminconsumer.PaintShopAdminConsumerActivator
11 Bundle-RequiredExecutionEnvironment: JavaSE-15
12
```

Figure 13: Admin consumer manifest

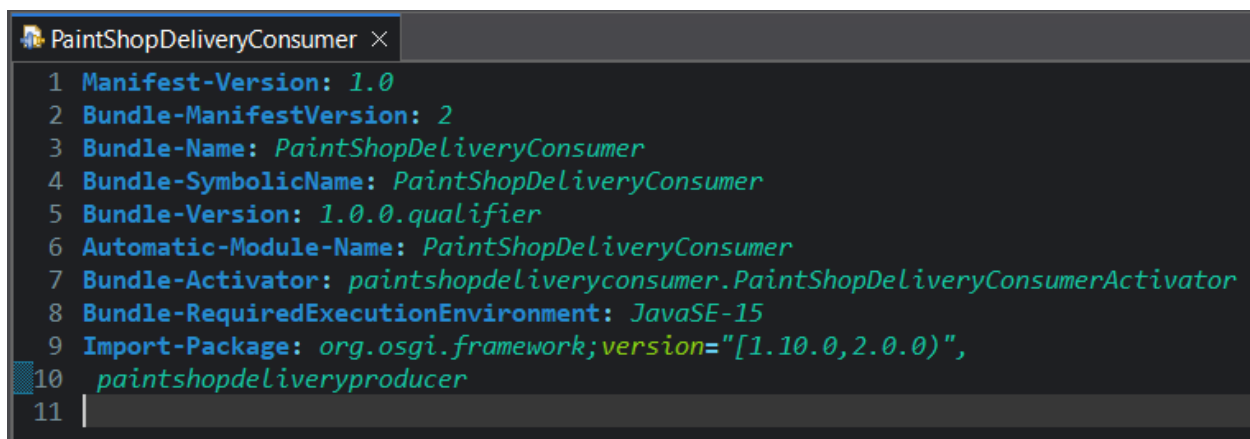
CUSTOMER CONSUMER



```
PaintShopCustomerConsumer ×
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: PaintShopCustomerConsumer
4 Bundle-SymbolicName: PaintShopCustomerConsumer
5 Bundle-Version: 1.0.0.qualifier
6 Automatic-Module-Name: PaintShopCustomerConsumer
7 Bundle-Activator: paintshopcustomerconsumer.PaintShopCustomerConsumerActivator
8 Bundle-RequiredExecutionEnvironment: JavaSE-15
9 Import-Package: org.osgi.framework;version="[1.10.0,2.0.0)",
10   paintshopcustomerproducer,
11   paintshopmixerproducer,
12   paintshoporderproducer
13
```

Figure 14: Customer consumer manifest

DELIVERY CONSUMER



```
PaintShopDeliveryConsumer ×
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: PaintShopDeliveryConsumer
4 Bundle-SymbolicName: PaintShopDeliveryConsumer
5 Bundle-Version: 1.0.0.qualifier
6 Automatic-Module-Name: PaintShopDeliveryConsumer
7 Bundle-Activator: paintshopdeliveryconsumer.PaintShopDeliveryConsumerActivator
8 Bundle-RequiredExecutionEnvironment: JavaSE-15
9 Import-Package: org.osgi.framework;version="[1.10.0,2.0.0)",
10   paintshopdeliveryproducer
11
```

Figure 15: Delivery consumer manifest

ALL COMMANDS OF BUNDLE INSTALL AND RUN IN OSGI FRAMEWORK

Make sure you have Equinox OSGi installed

Running the OSGi Bundles Without Exporting JARs (Directly from Eclipse)

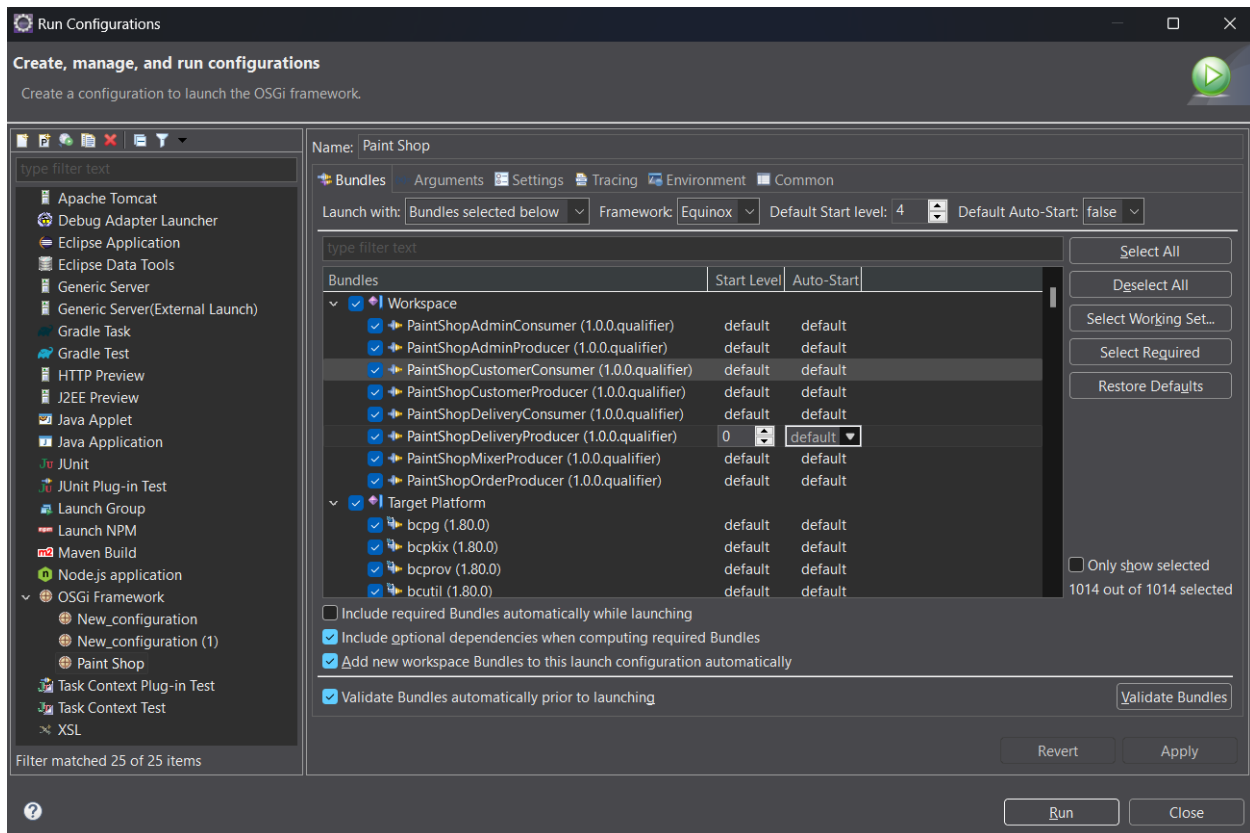


Figure 16: Run Configurations

Step 1: Open "Run Configurations" in Eclipse

1. Click on Run → Run Configurations.
2. Select OSGi Framework from the left panel.
3. Ensure all required bundles are selected (as shown in screenshot).

Step 2: Configure the OSGi Framework

1. Framework: Ensure Equinox is selected.
2. Auto-Start: Set false for all Producer and Consumer bundles.
3. Validate Bundles: Click on Validate Bundles to ensure dependencies are resolved

Step 3: Run the OSGi Framework

1. Click Apply → Run.
2. Eclipse will start the OSGi framework and execute the bundles.
3. Open the Eclipse Console to view logs/output

Step 4: Verify Running Bundles

1. After launching, Open the OSGi console in Eclipse.
2. Type: lb or ss

```
Welcome to Apache Felix Gogo

g! lb
START LEVEL 6
  ID|State      |Level|Name
  0|Active      |    0|OSGi System Bundle (3.23.0.v20250228-0640)|3.23.0.v20250228-0640
  1|Active      |    1|Simple Configurator (1.5.400.v20250129-0942)|1.5.400.v20250129-0942
  2|Resolved    |    4|PaintShopAdminConsumer (1.0.0.qualifier)|1.0.0.qualifier
  3|Starting    |    4|PaintShopAdminProducer (1.0.0.qualifier)|1.0.0.qualifier
  4|Resolved    |    4|PaintShopCustomerConsumer (1.0.0.qualifier)|1.0.0.qualifier
  5|Starting    |    4|PaintShopCustomerProducer (1.0.0.qualifier)|1.0.0.qualifier
  6|Resolved    |    4|PaintShopDeliveryConsumer (1.0.0.qualifier)|1.0.0.qualifier
  7|Starting    |    4|PaintShopDeliveryProducer (1.0.0.qualifier)|1.0.0.qualifier
  8|Starting    |    4|PaintShopMixerProducer (1.0.0.qualifier)|1.0.0.qualifier
  9|Starting    |    4|PaintShopOrderProducer (1.0.0.qualifier)|1.0.0.qualifier
 10|Resolved    |    4|bcpq (1.80.0)|1.80.0
 11|Resolved    |    4|bcnkix (1.80.0)|1.80.0
```

Figure 17: Verify Bundles

Admin Workflow

1. To Run Type: start 2
2. To Stop Type: stop 2

```
g! start 2
Start Admin Consumer Service
PaintShop Admin Producer Started
Paint Shop Mixer Producer Started
_____ Paint Shop Admin Dashboard _____

Choose an option:
1. Add New Paint Mixer
2. Add a Delivery User
3. Exit
Enter your choice:
```

Figure 18: Admin Workflow Output SS

```
Enter your choice: 1

--- Add New Paint ---
Enter Color Code: #ff0000
Enter Paint Name: red
Enter Paint Price: 100
New Paint added successfully!
Paint added successfully!

Choose an option:
1. Add New Paint Mixer
2. Add a Delivery User
3. Exit
Enter your choice:
```

Figure 19: Admin Workflow Output SS


```
3. Exit
Enter your choice: 2

--- Register New Delivery User ---
Enter User Name: Driver 2
Enter Phone Number: 123456
Enter Username: d2
Enter Password: 123
Delivery user already registered!
Driver 2 123456

Choose an option:
1. Add New Paint Mixer
2. Add a Delivery User
3. Exit
Enter your choice:
```

Figure 20: Admin Workflow Output SS

```
Choose an option:
1. Add New Paint Mixer
2. Add a Delivery User
3. Exit
Enter your choice: 3
Exiting Admin Consumer...
g!
```

Figure 21: Admin Workflow Output SS

Customer Workflow

1. To Run Type: start 4
2. To Stop Type: stop 4

```
Start Customer Consumer Service
Paint Shop Customer Producer Started
Paint Shop Order Producer Started
_____ Welcome to the Paint Shop! _____
Do you have an account? (Y/N)
n
--- Paint Shop Customer Registration ---
Enter Your Name: cus 1
Enter Your Phone Number: 123456
Enter Your Email: cus1@gmail.com
Enter Your Address: colombo
Registration successful! Welcome cus 1
Do you have an account? (Y/N)
y
--- Paint Shop Customer Login ---
Enter Your Name: cus 1
Enter Your Phone Number: 123456
Hi cus 1, welcome back!
Do you want to buy paint? (Y/N)
```

Figure 22: Customer Workflow Output SS

```
Do you want to buy paint? (Y/N)
y
|
--- Available Paints in Shop ---

White!#0000!100
black!#ffff!100
red!#ff0000!100
Choose a Paint by color code:
```

Figure 23: Customer Workflow Output SS

```

Choose a Paint by color code: #ff0000
New bill file created: cus 1616763.txt
Successfully Inserted!!!
Do you want to buy paint? (Y/N)
n

--- Bill Details ---
Color Code: #ff0000, Quantity: 1, Price: 100
Do you want to deliver? (Y/N)
y
Successfully inserted into Bill Register!
g!

```

Figure 24: Customer Workflow Output SS

Driver Workflow

1. To Run Type: start 6
2. To Stop Type: stop 6

```

--- Paint Shop Driver Login ---
Enter Your Username: amila
Enter Your Password: 123
Hi amila, welcome!

-----|
----- Delivery Packages -----|
-----|

Bill ID | Price | Status |
cus820454 | 100 | YES |
cus 1616763 | 100 | YES |
g!

```

Figure 25: Driver Workflow Output SS

Running the Exported OSGi JAR Bundles

Step 1: Open Terminal / Command Prompt and Navigate to the Directory

Type: `cd /path/to/JAR`

Replace with the actual path, where you saved your JAR files.

Step 2: Start the OSGi Framework

Run the Equinox OSGi framework launcher:

Type: `java -jar org.eclipse.osgi_VERSION.jar -console`

Make sure to replace `org.eclipse.osgi_VERSION.jar` with the actual name of your Equinox JAR.

Step 3: Install All Bundles

Inside the OSGi console, install each JAR file:

Type:

- `install file:Admin\ Consumer.jar`
- `install file:Admin\ Producer.jar`
- `install file:Customer\ Consumer.jar`
- `install file:Customer\ Producer.jar`
- `install file:Delivery\ Consumer.jar`
- `install file:Delivery\ Producer.jar`
- `install file:Mixer\ Producer.jar`
- `install file:Order\ Producer.jar`

Step 4: Continue from [This](#)