Software Architecture

Assignment 1

Micro-Kernel Architecture & OSGi framework

Report

Group Members

|  |  |
| --- | --- |
| Student ID | Student Name |
| IT18053838 | H.P Sumudu Lakruwan |
| IT18029246 | Saranga S.A.N |
| IT18027884 | Fernando G.U. S |
| IT18020618 | Somarathne R.P.A. B. |

Individual Contributions

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student Name** | **Individual Contributions** |
| IT18053838 | H.P Sumudu Lakruwan | * MedicineHelperManager implementation * MedicineHelperManager manifest implementation * MedicineManagementVersion1 implementation and manifest implementation * MedicineManagementVersion2 implementation and manifest implementation * MainConsumer Implementation(integrate part) and and subscriber implementation for the MedicineManagementVersion1 and MedicineManagementVersion2 * High level diagram designing * Contributions for the final report |
| IT18029246 | Saranga S.A.N | * CustomerManage Producer implementation * CustomerManage function implementation in the MainConsumer subscriber i * Manifest implementation of CustomerManage Producer * High level diagram designing |
| IT18027884 | Fernando G.U. S | * Essential Products Management Producer implementation * MainConsumer subscriber implementation for the Essential Products Management * Manifest implementation of Essential Products Management Producer * Manifest implementation of MainConsumer consumer * High level diagram designing   Contributions for the final |
| IT18020618 | Somarathne R.P.A. B. | * Delivery Management Producer implementation * MainConsumer subscriber implementation for the Delivery Management * Manifest implementation of Delivery Management Producer * Manifest implementation of MainConsumer consumer * Abstraction and Scenario/Introduction   Contributions for the final |

Abstract

This report describes briefly about the scenario, implementations, components and functionalities about the designed application. Further screenshots and diagrams are used to give more understanding about the application.

Scenario/Introduction

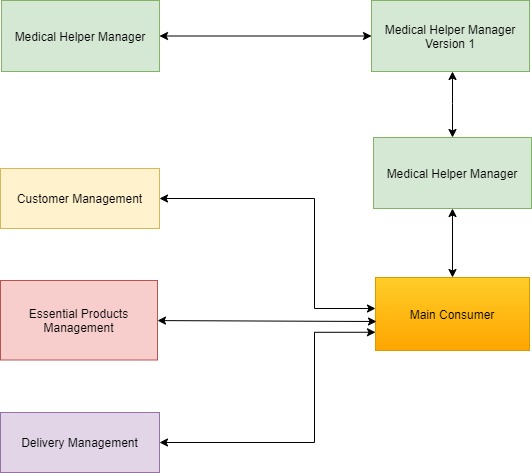
The Urgent Service Management system is divided into 4 main parts and those parts are working according to OSGi producer and consumer bundles where consumer can get the services from producer and once done consumer can unregister from services. Producer bundles calculate, manage and output the results to the consumer.

The systm is designed for mange urgent services for the customer.We mange custormer urgent requirement such as urgent medicines,foods and delivary services.

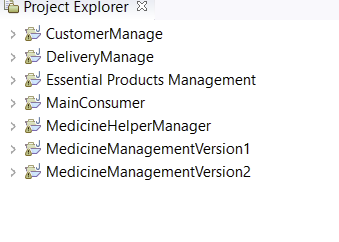
When there is corona disaster or something happening,this system is designed for such situations.For that we use MedicineManagementVersion1, MedicineManagementVersion2 and MedicineHelperManager bundles for urgent medicines managenment.

CustormerMange bundle to mange custormer.DelivaryMange bundle is used for manage delivary.Essential Products management bundle is used for .Essential Products management.MainConsumer bundle is use for consuming

System Diagram

****

Project view of the whole system

Whole system project structure

IT18053838 H.P Sumudu Lakruwan

(1)MedicineManagementVersion1

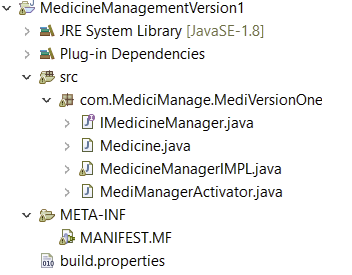


Figure - MedicineHelperManagerVersion1 project structure

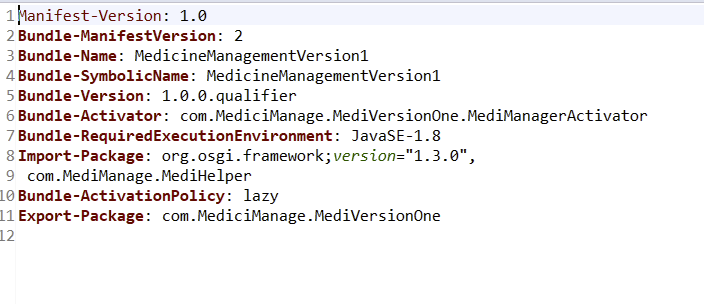
This helps to mange urgent medicine details.In order to manage urgent medicines details,I used MedicineManagementVersion1 and MedicineManagementVersion2 . MedicineManagementVersion1 is service publisher to MainConsumer and serviceConsumer for the MedicineHelperManager. This MedicineManagementVersion1 has funtions such as

(1)Add Medicine details – medicine id,medicine type,medicine quantity,medicine unit price

(2)Display medicine details using medicine id

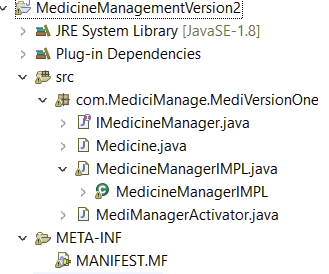
(3)Delete medicine and more.

Manifest file



(2) MedicineManagementVersion2

Project stucture



This has all functions same as the MedicineManagementVersion1 has.But This MedicineManagementVersion2 has extra detalils called Medicine decription inside add method

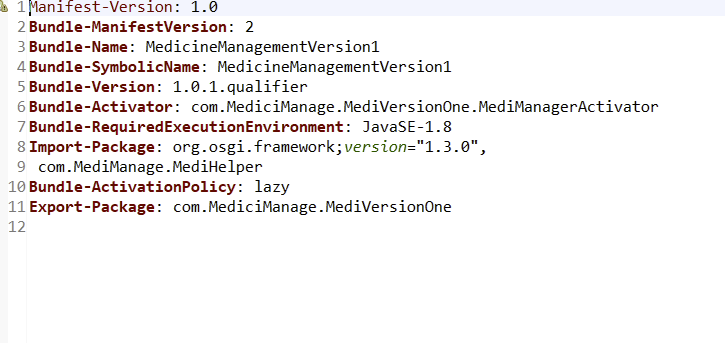
1)Add Medicine details – medicine id,medicine type……..,MedicineDescription

(2)Display medicine details using medicine id

(3)Delete medicine and more.

MedicineManagementVersion2 is publisher to MainConsumer and serviceConsumer for the MedicineHelperManager

**Manifest**

****

(3) MedicineHelperManager

This is service publisher to both MedicineManagementVersion1 and MedicineManagementVersion2.We have medicine type A,B,C and D.In the store we can store only certain amount of quantity for each type.

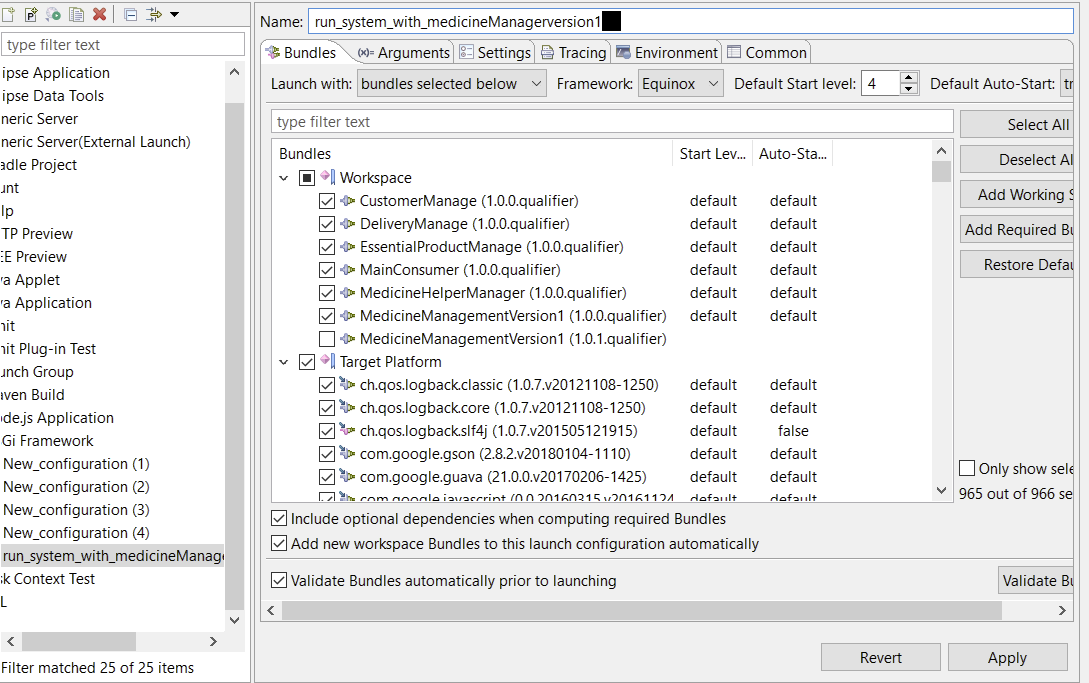
Ex:Type A- can be store less than 250, Type B-can be store less than 350

When user enter quantity for the given type,this Medicinehelper check whether we can add quantity to given type in the store.

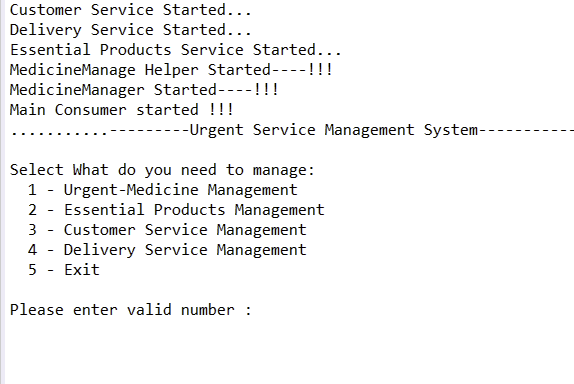
How MedicineHelperManagerVersion1, MedicineHelperManagerVersion2 and Medicinehelper works

**Option 1- MedicineManagementVersion1**

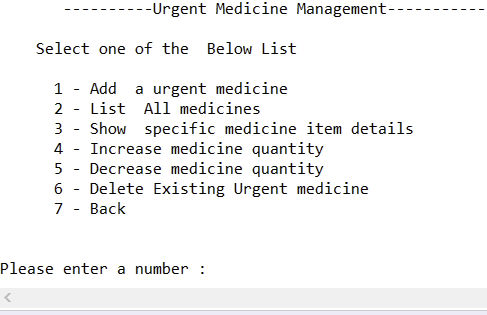
(1)go to below window select MedicineManagementVersion1 with other bundles except MedicineManagementVersion2.

****

(2)Then you can see below screen in eclipse terminal

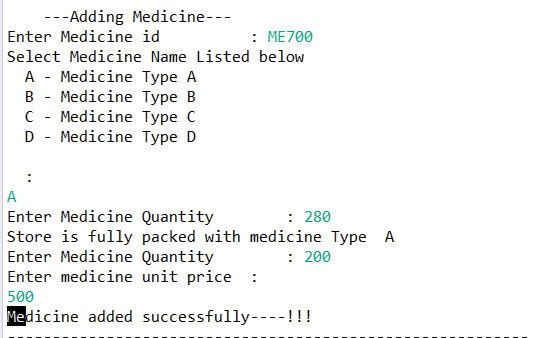
****

**(3)** (3)Then select Urgent-meicine management by entering number 1,then you can seen below in eclipse terminal

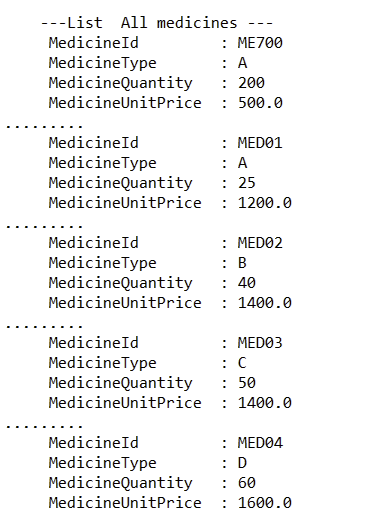


(4)In there you can add urgent medicine,List all medicine,show specific medicine item,increase medicine quantity,decrase quantity and more things can be done.

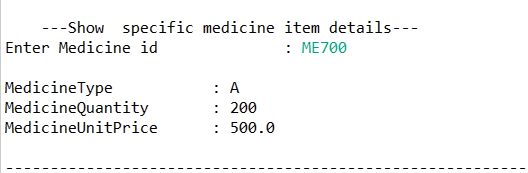
* Add Urgent medicine

****

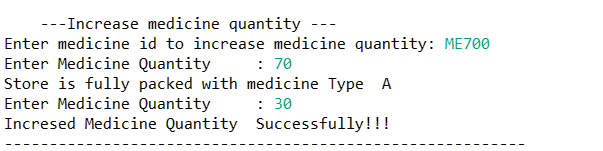
* List all medicines



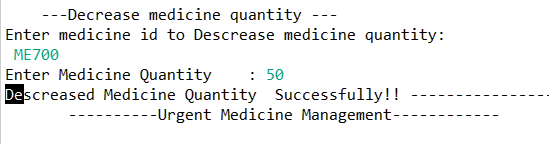
* Show specific medicine details



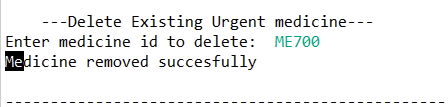
* Increase quantity(incrase quantity by adding new quantity to existing quantity)-Medicinehelper check whether quantity can increase or no.



* Decrease medicine quantity



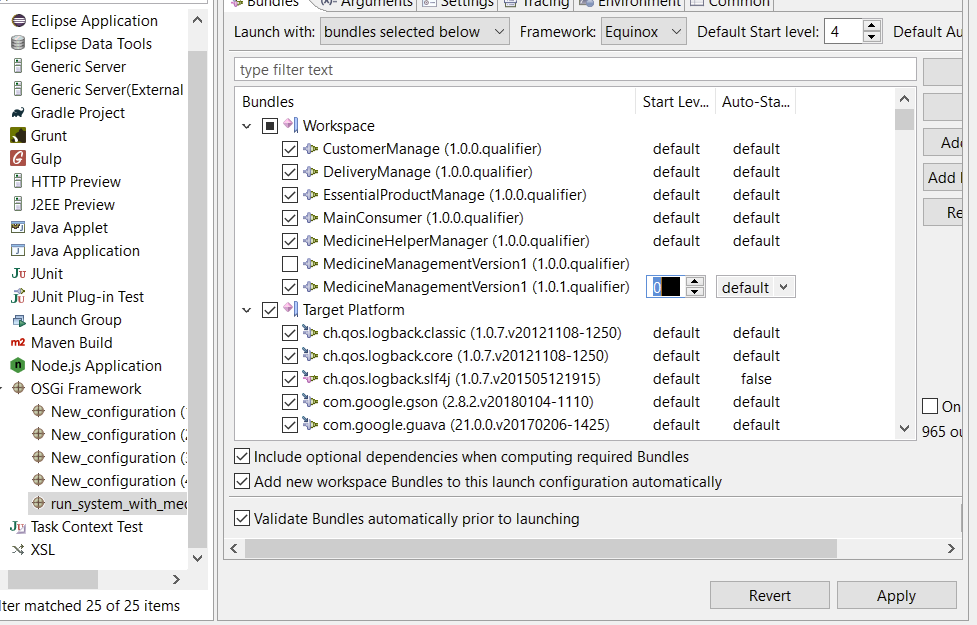
* Delete existing urgent medicine



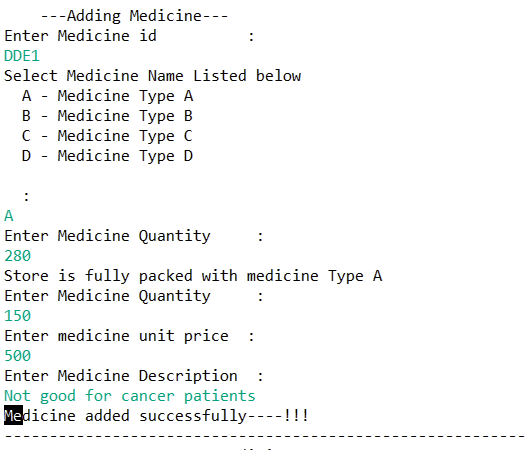
**Option 2- MedicineManagementVersion2**

In here only different is we can add urgent medicine description inside the add method

**(1)** go to below window select MedicineManagementVersion2 with other bundles except MedicineManagementVersion1

****

* Add urgent medicine



IT18027884 - Fernando G.U. S

1. **Essential Products Management**

Detailed flow of the function

Form the Essential product management main menu users can add, view, update and delete the products in budget packs. System has 3 Budget packs,

1.Budget Pack 1500 Rs

2.Budget Pack 2500 Rs

3.Budget Pack 7500 Rs

1. **Add Products to Budget Packs**

First user should select a budget pack then user should provide a unique item id then it will check if the id is unique if not it will ask for a unique id I the id is unique then user can provide product name and price then it will be added to the relevant Array List.

1. **Show Products**

This function will display all the budget packs and its product details.

1. **Update Products**

First user should select a budget pack then user should provide item id of the product that wants to update. Then name and price can be changed. If the provided id is in the system update will success if not update will fail.

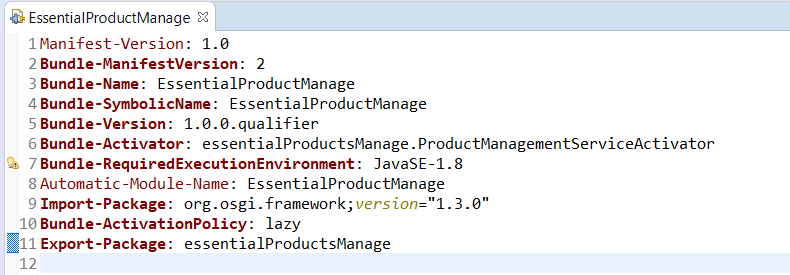
1. **Delete Products**

First user should select a budget pack then user should provide item id of the product that wants to delete. If the provided id is in the system item will be deleted.

**2.Manifest Implementation**

1. Essential Products Management

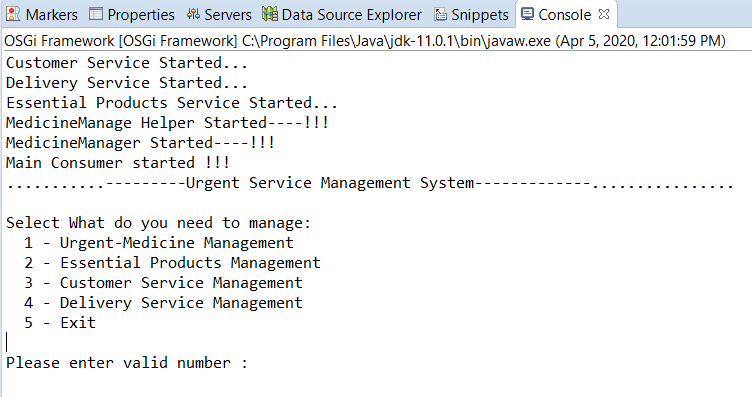
(Producer)

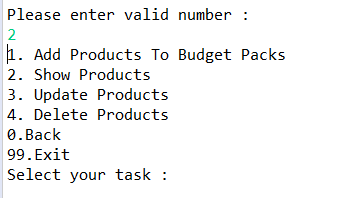


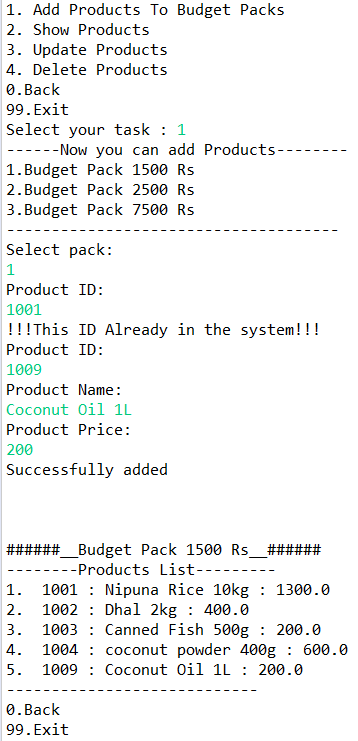
(Essential Products Management Manifest)

**3.Screenshots**

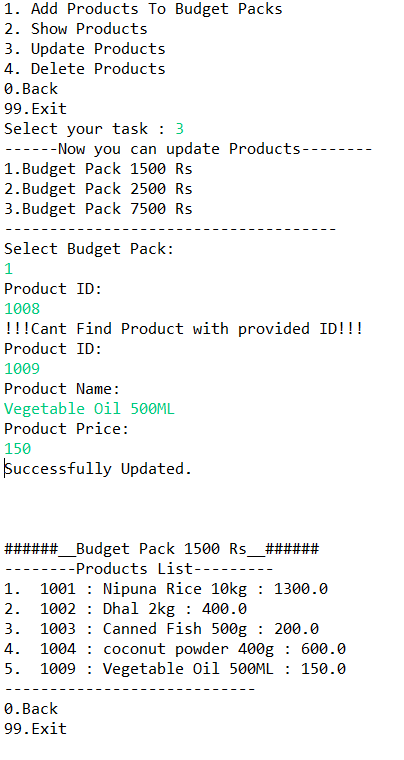
1. **Essential Products Management**



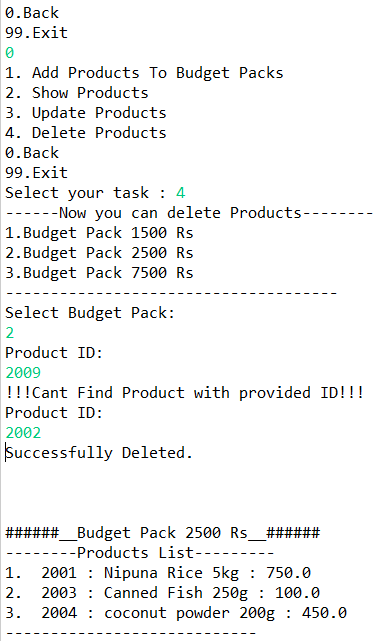




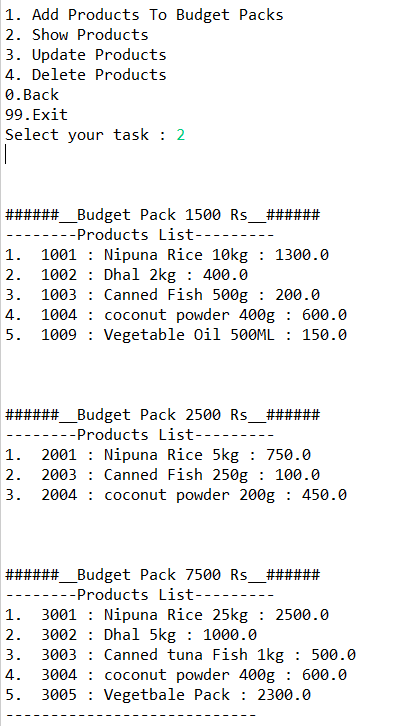
(Adding Products)



Updating Products)



(Deleting Products)



(View Products)

**IT18029246 - Saranga S.A.N**

**Customer registration**

Flow of the customer registration.

1. Initially shows the main menu.
2. Select no.3 from the main menu (Customer Service Management).
3. Then it'll display two categories which are 'Add New Customer' and 'Show All Customers' from Customer Manage Menu.

**Add New Customer**

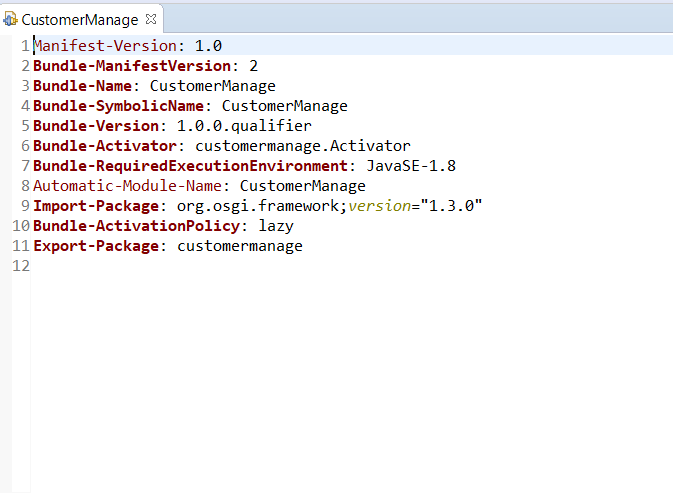
1. If you select no.1 from Customer Manage Menu, it'll go to Customer Registration part.
2. From ‘Select Customer Type’ menu you can select either ‘Elders who receive Pensions’ by selecting no.1 or ‘Other customers’ by selecting no.2
3. It asks to fill some information such as customer ID, customer Name, customer phone No and customer address.

**Show All Customers**

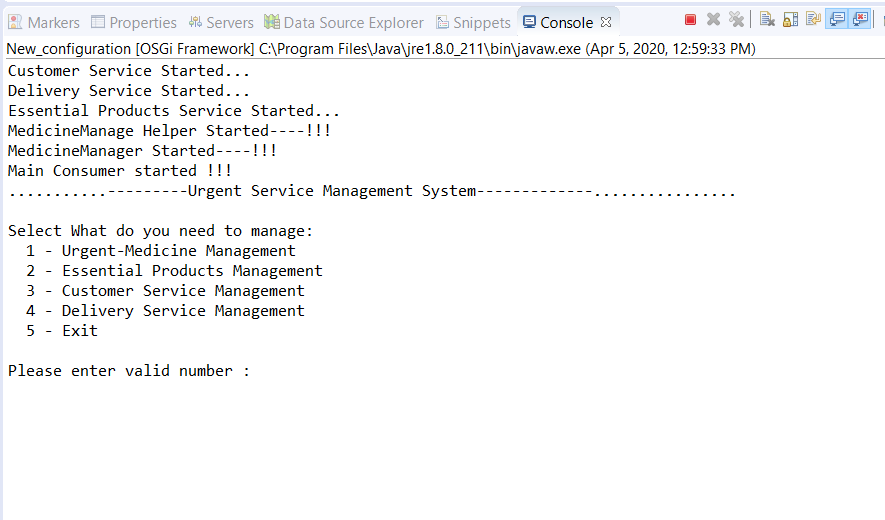
1. If you select no.2 from Customer Manage Menu all the registered customers details will be displayed.

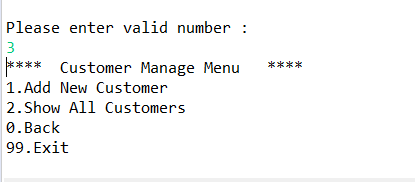
Further you can move back by selecting number 0 and directly to the main menu by selecting number 99.

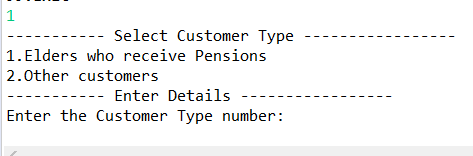
**Customer registration**

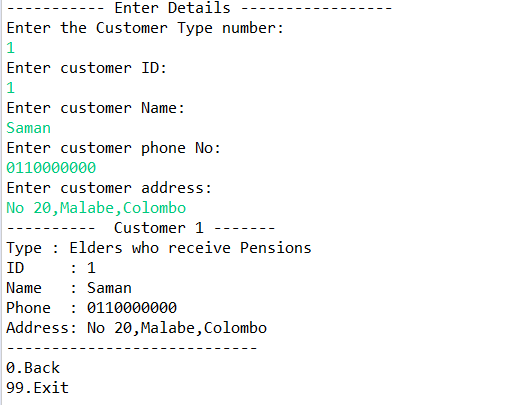


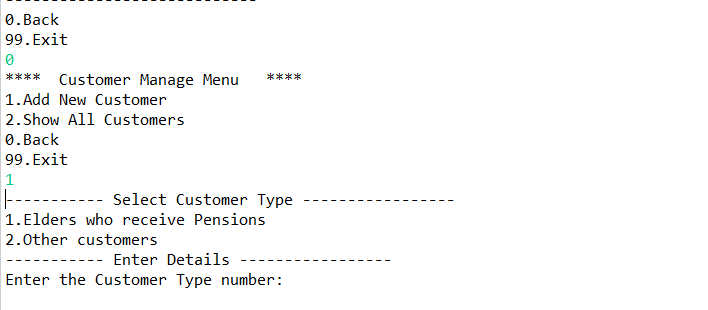
**Customer registration**

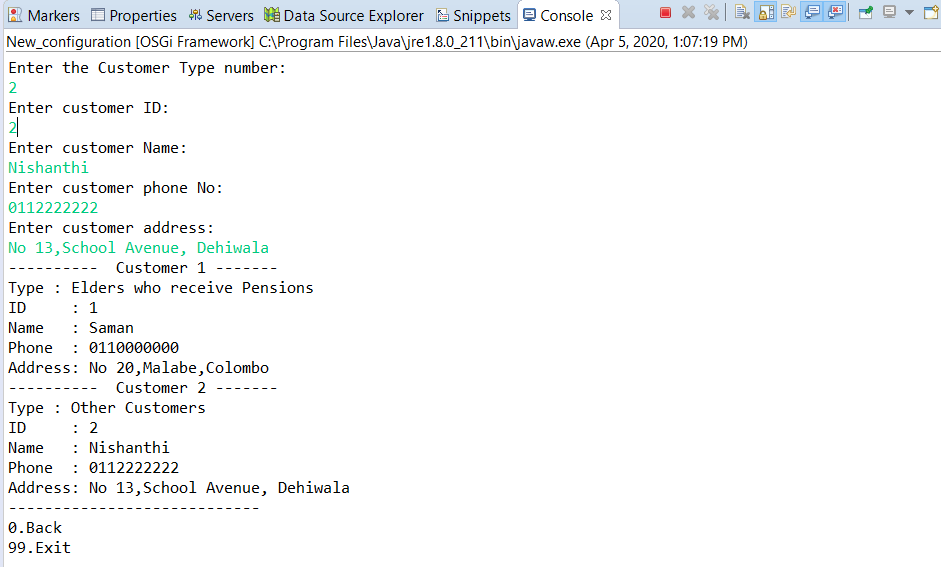


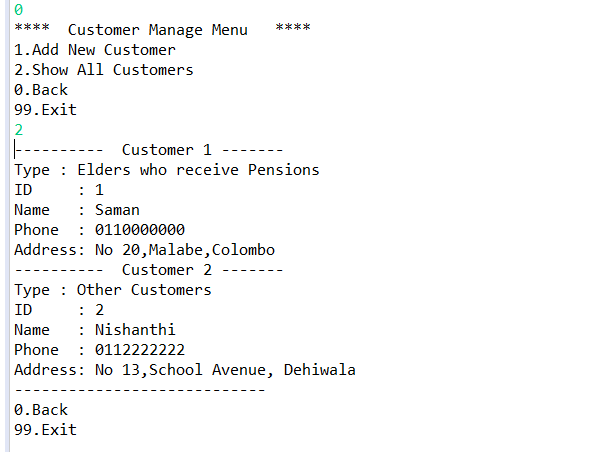












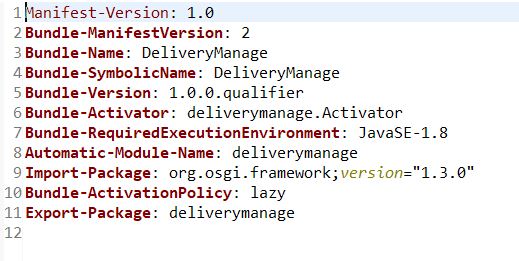
IT18020618- Somarathne R.P.A. B

1. **Delivery Management**

Detailed flow of the function.

1. After starting the program the main menu which is consisted of sub menus will be prompted to the screen.
2. After typing the number 2, Delivery manage menu will be prompted which has other two sub menus.
3. By typing number 1, user can assign a new delivery for a customer. A set of areas will be displayed in the screen. User must select one of them and next user must type driver’s vehicle number, driver’s name and customer id to assign a new delivery.
4. By typing the number 2 in the delivery manage menu, user can view all the deliveries that are assigned to customer with details.
5. Further user can move back and directly to the main by giving number 0 and number 99 respectively.
6. **Manifest Implementation**
7. Delivery Management

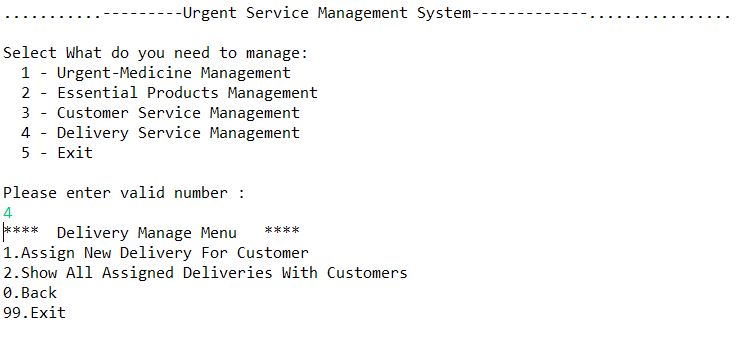
(Producer)

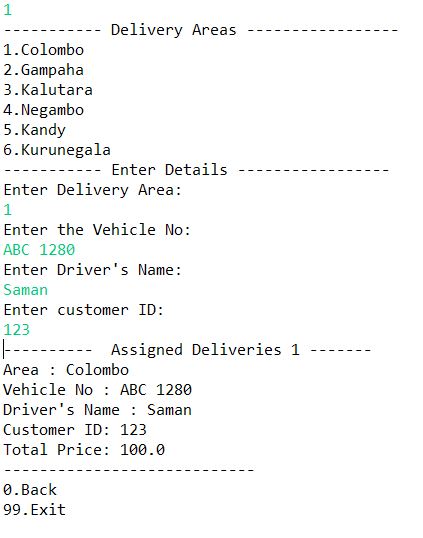


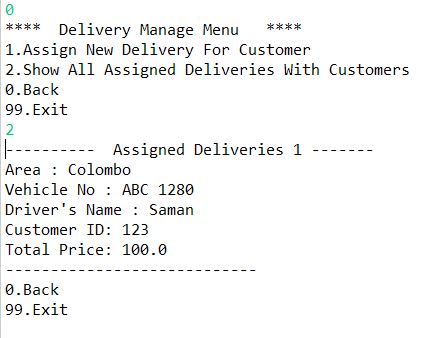
(DeliveryManage manifest)

1. **Screenshots**

**Delivery Management**



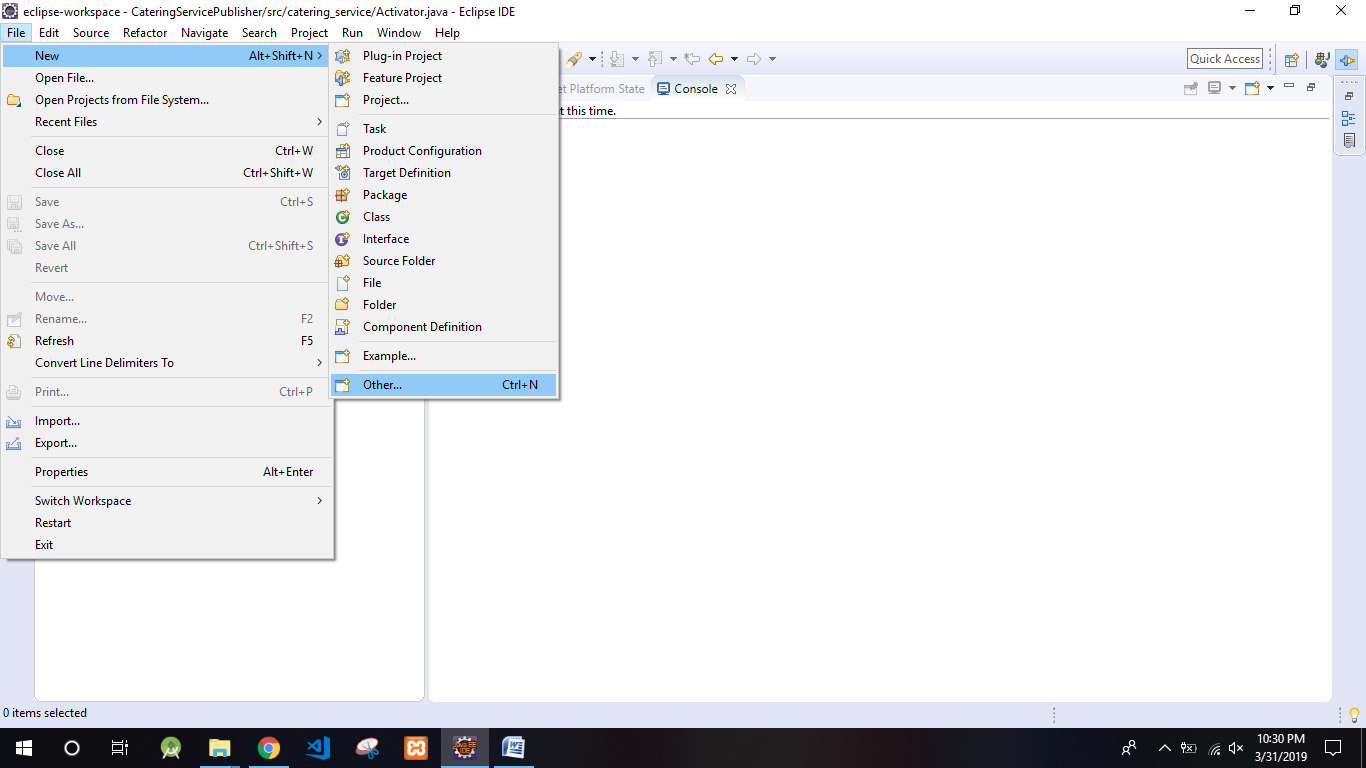


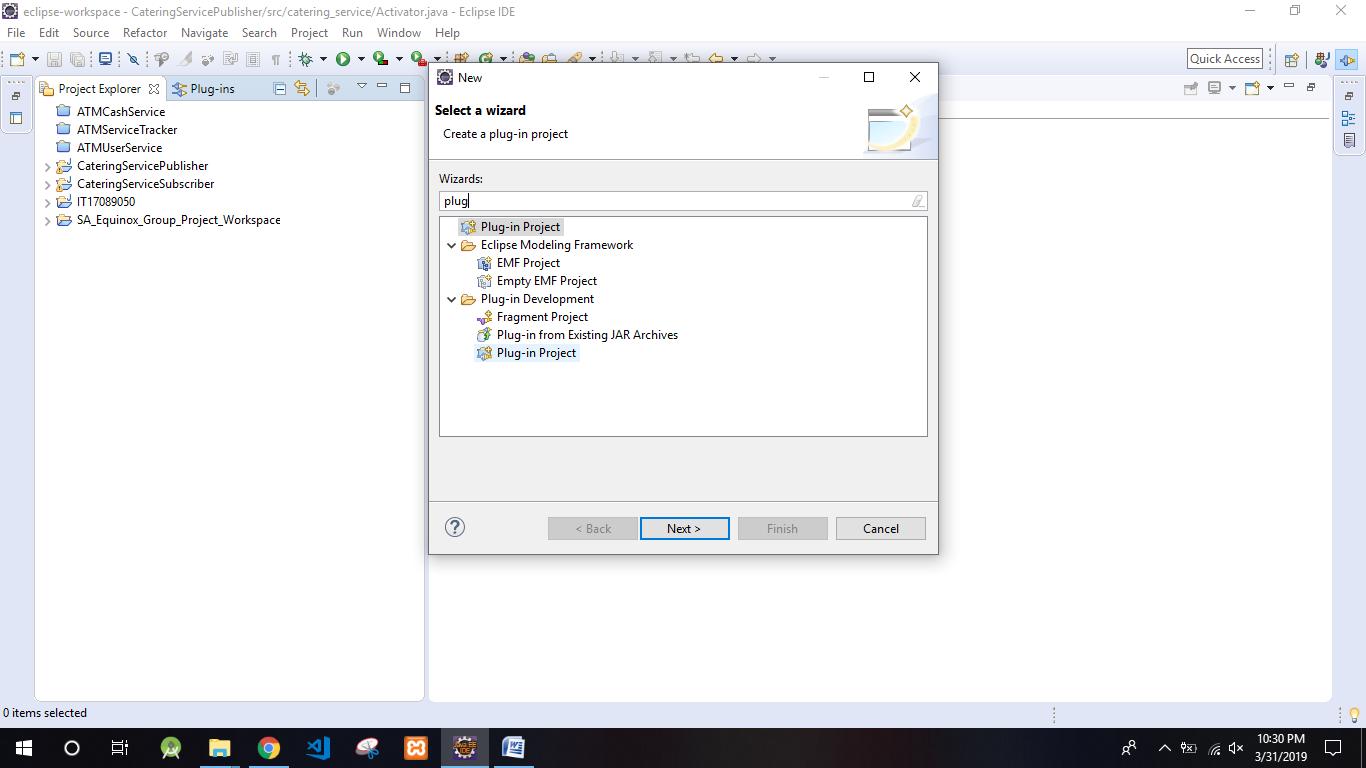


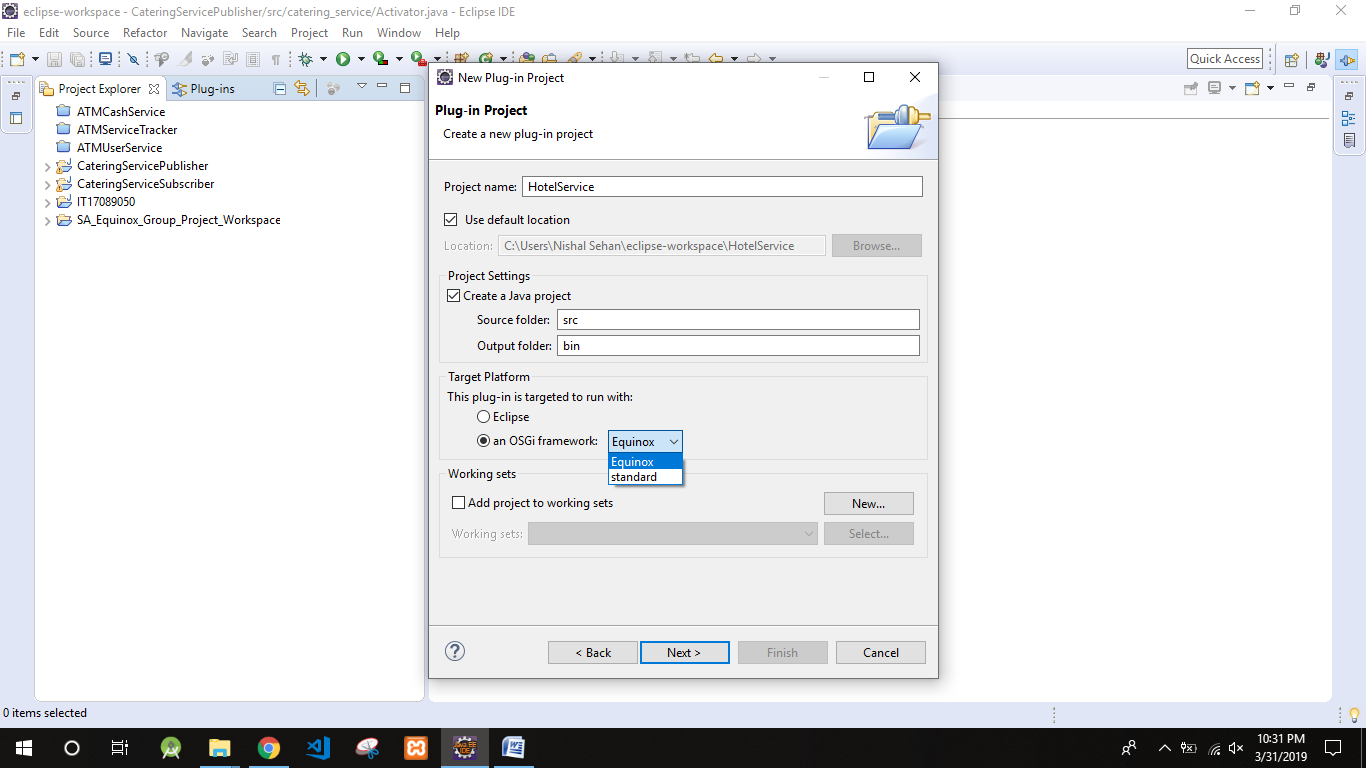
1. **Project implementation procedure**

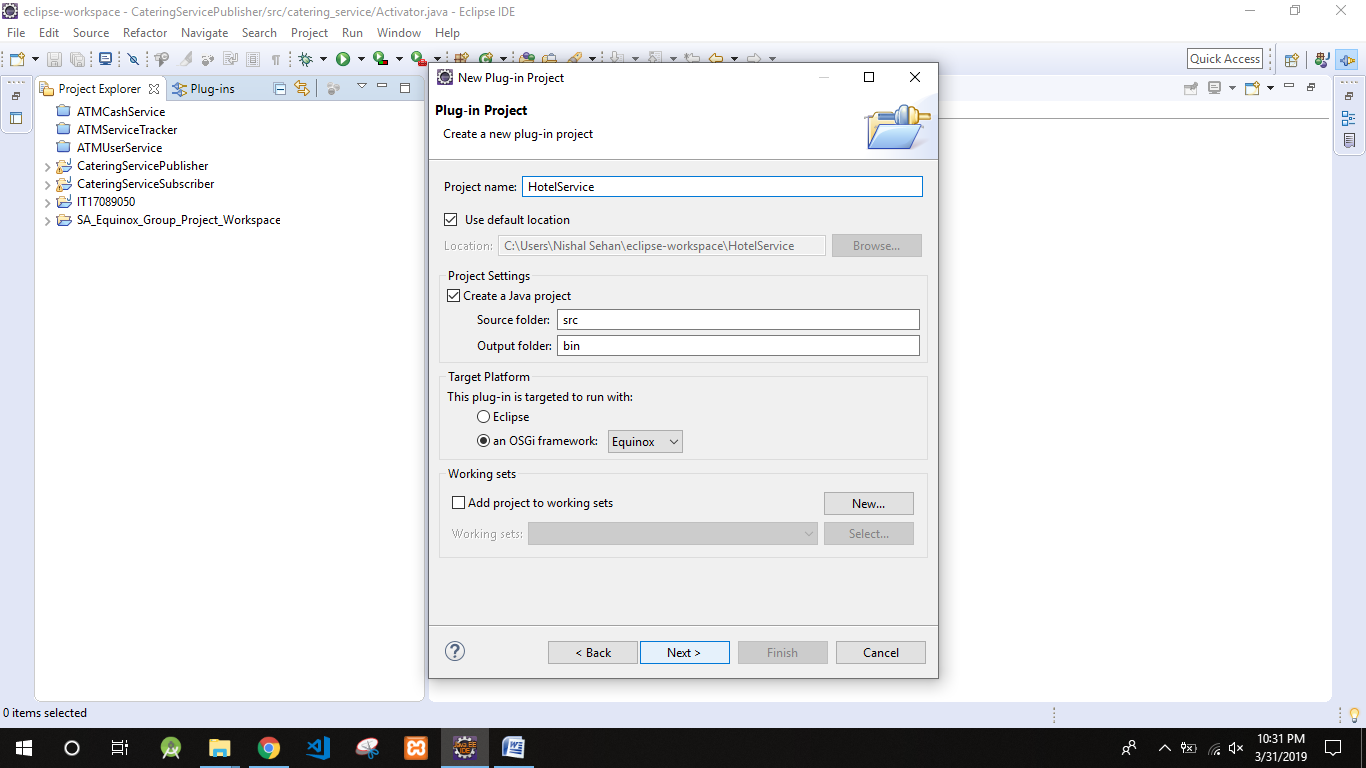
**Producer Implementation**

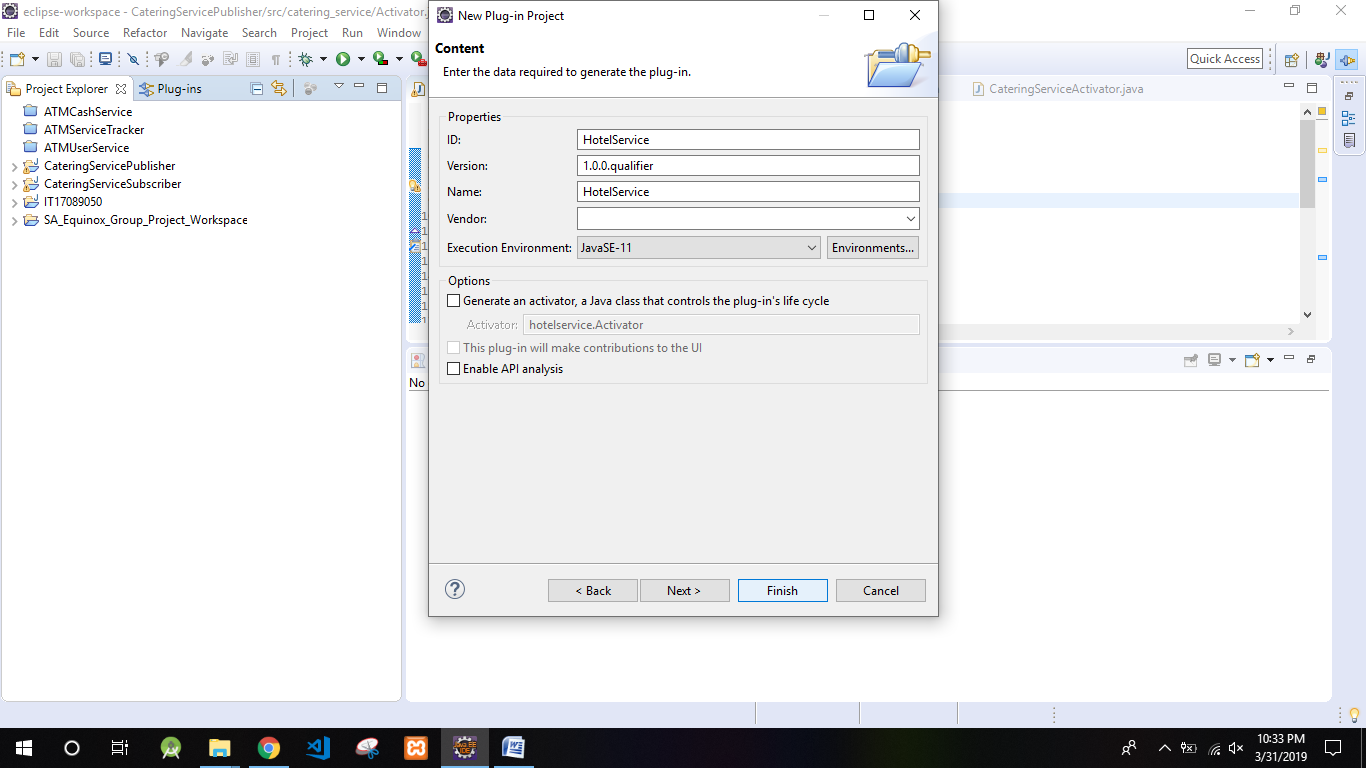
1. Create a ‘Plug-in Project’ from File > New > Other

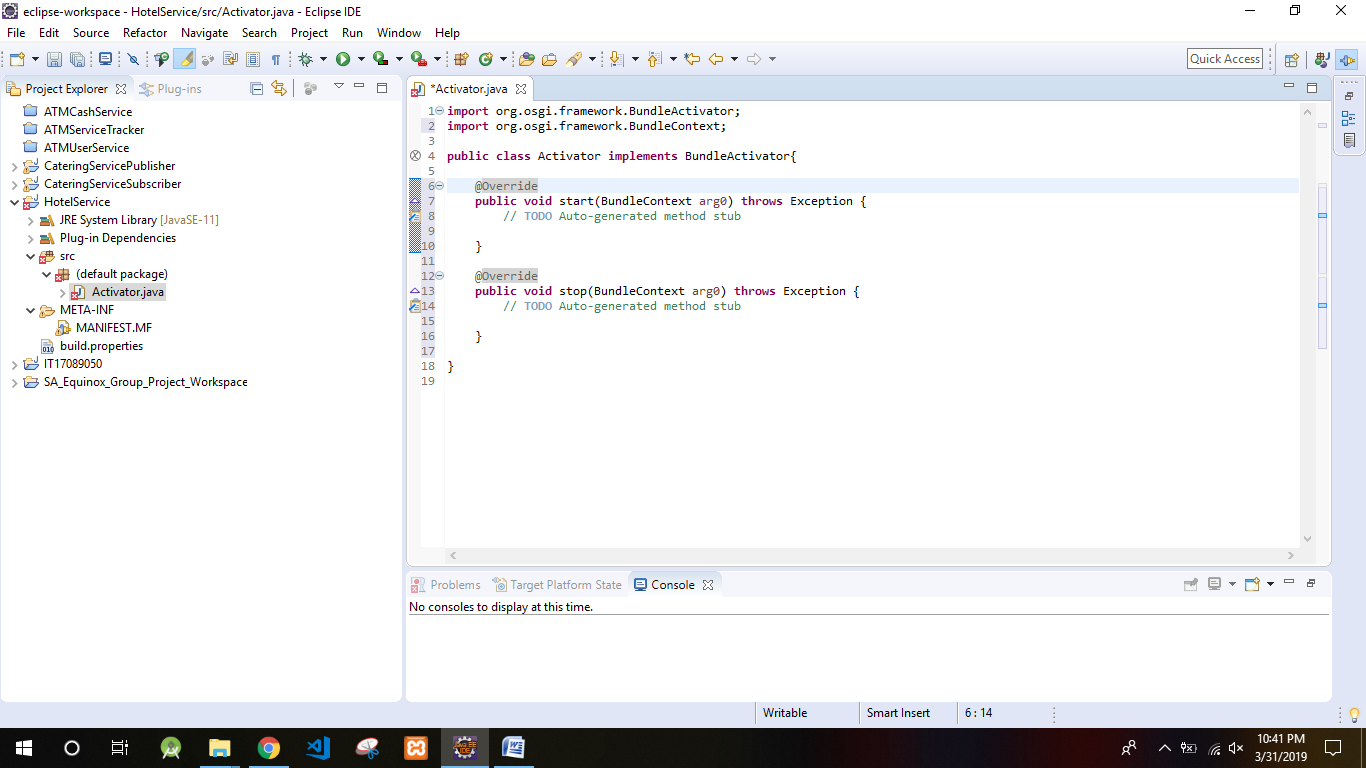






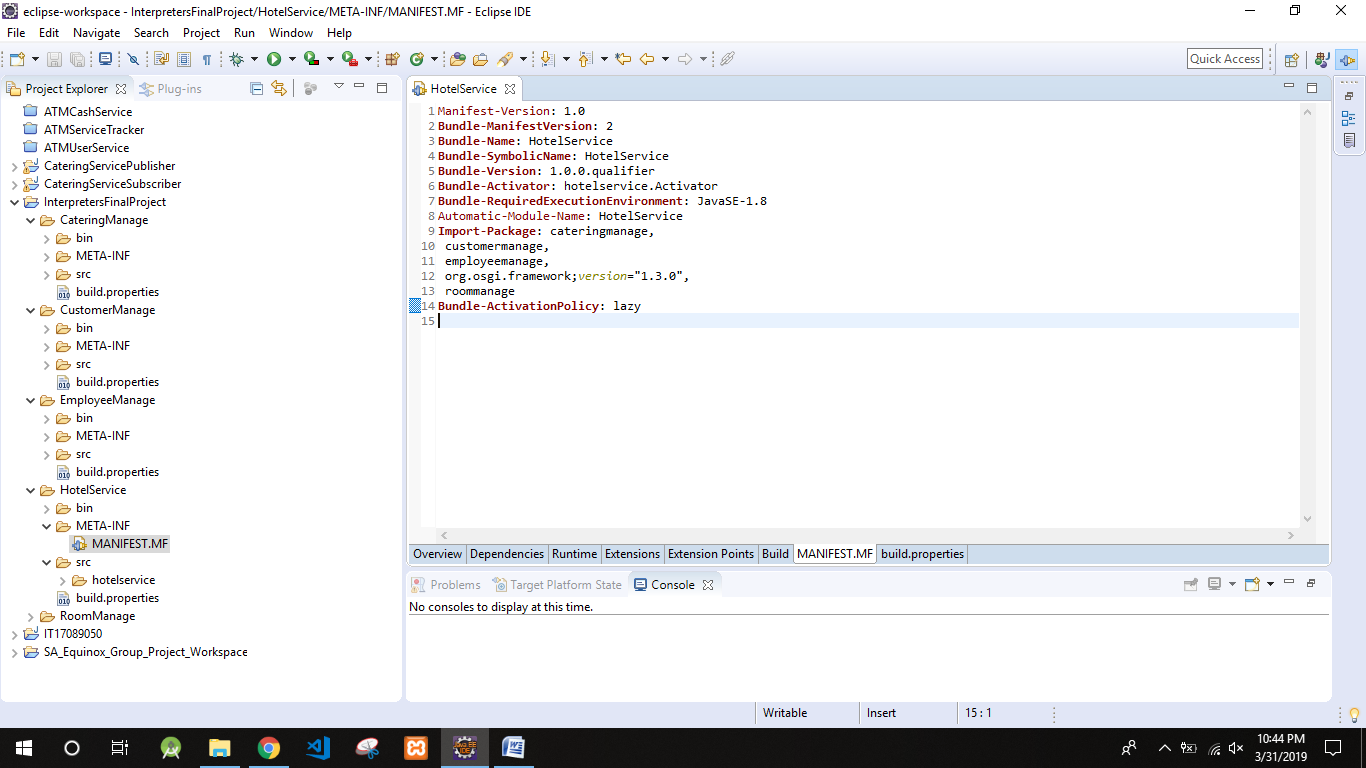






1. Access the ‘MANIFEST.MF’ file to declare the packages that need to be exported from the Producer.

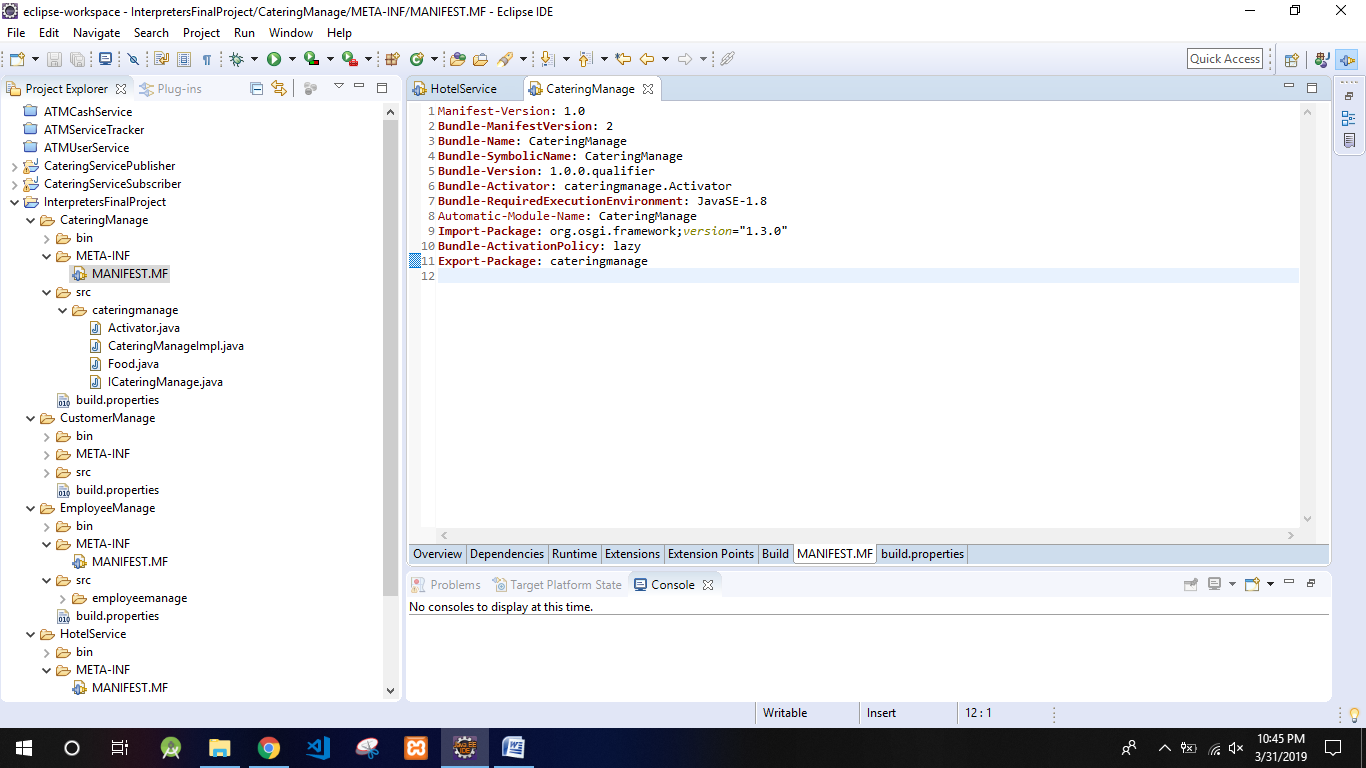
(‘MANIFEST.MF’ is in Project Directory > META-INF)



**Consumer Implementation**

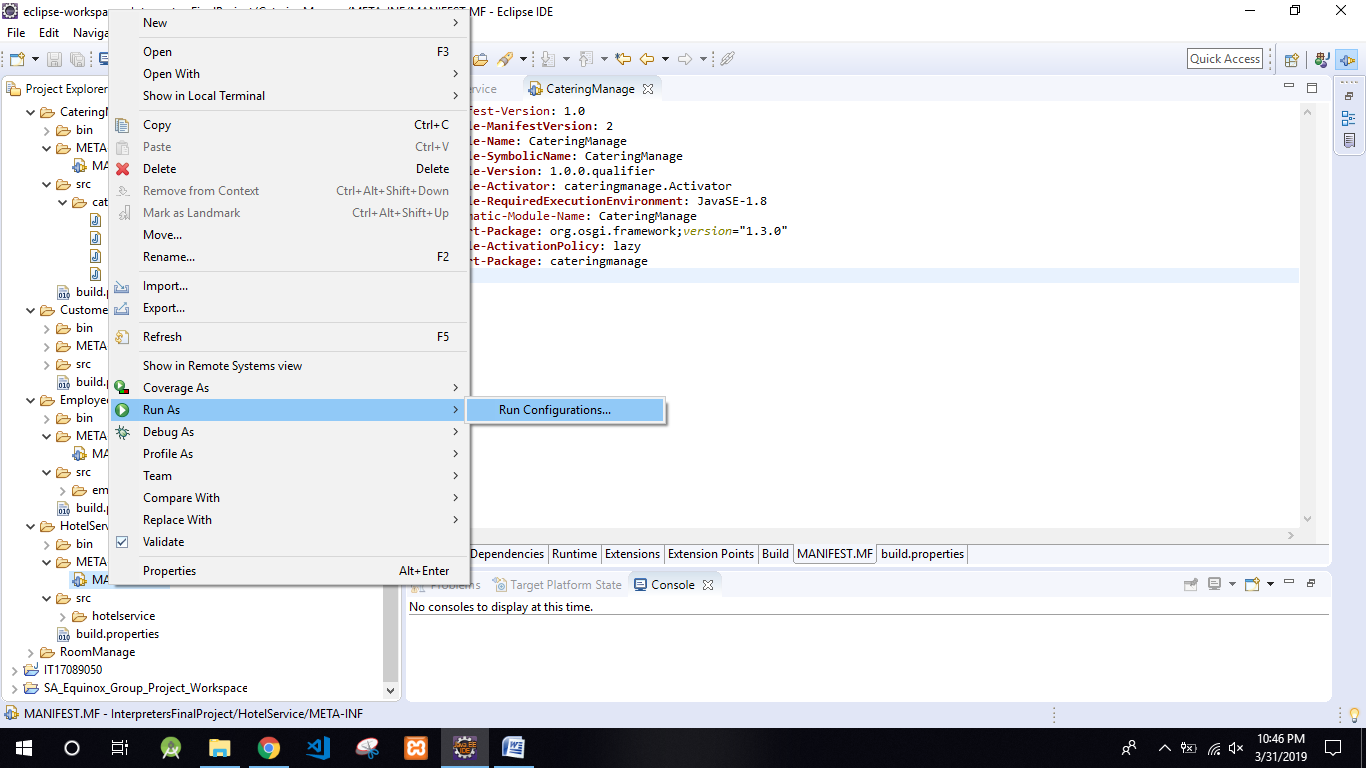
Repeat step 1 as in Producer to create a project.

1. Import packages that have exported from Producers’ ‘MANIFEST.MF’ to the Consumers ‘MANIFEST.MF’

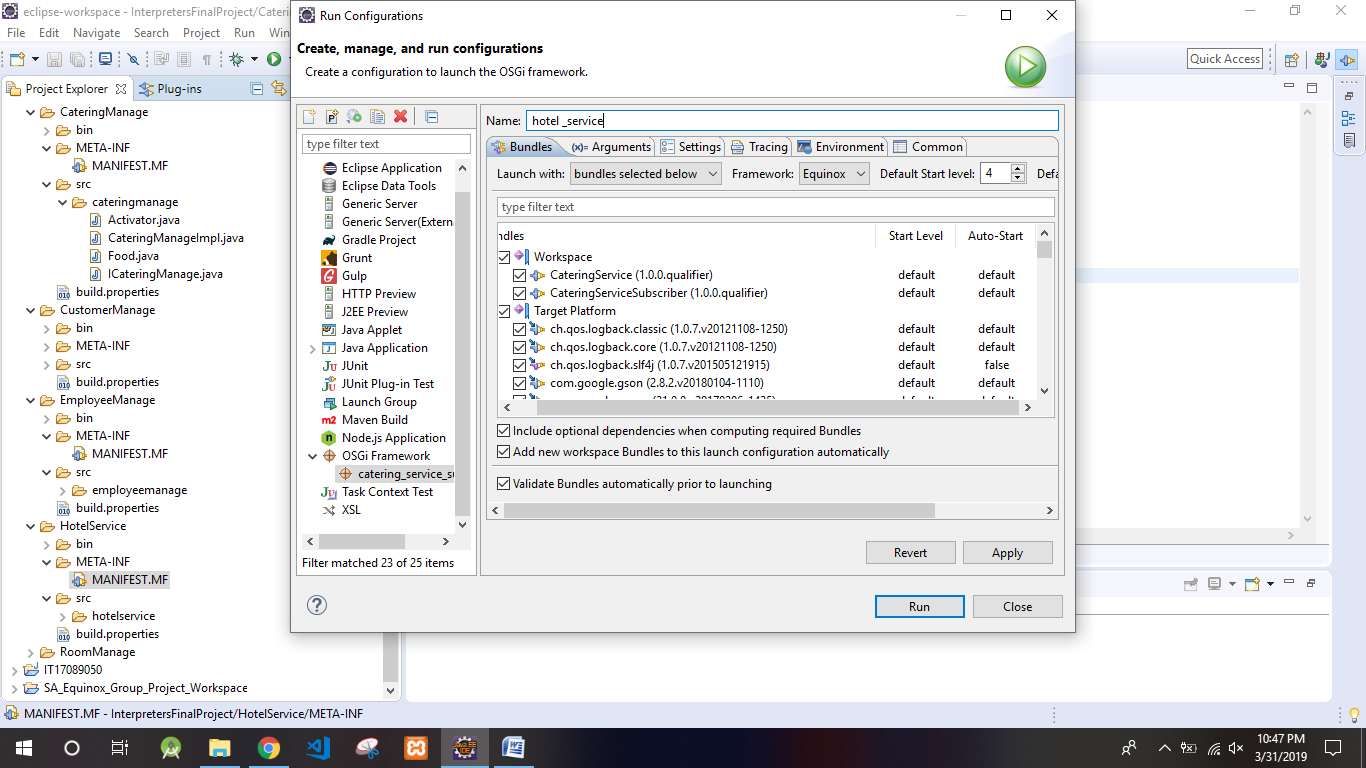


1. Create a separate run configuration to run Producer and Subscriber

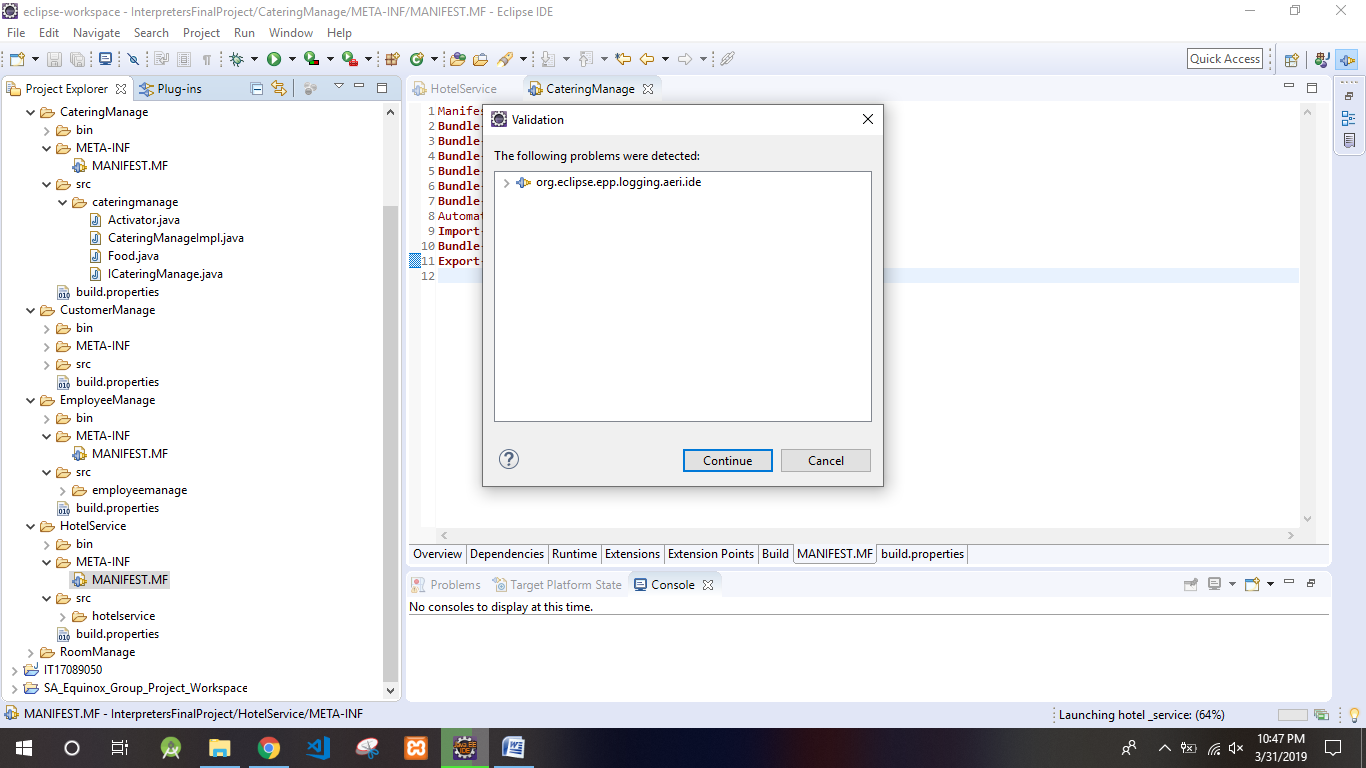
To create Right-click on MANIFEST.MF > Run as > Run Configurations



3.Select OSGi Framework and Click on ‘New’ to create new configuration



4.Add new name > Click ‘Run’



5.Finally run project by Right-Click on MANIFEST.MF > run as > OSGi Framework

6.Select the configuration you made and click ‘OK’

You can use OSGi commands to start stop and view details about bundles.

To view all bundle data use ‘lb’ (‘lb’ stands for list bundles).

To start and stop use ‘start <Bundle ID>’