Abstract

This document gives a brief explanation on the components, functionality and the environment of the designed application. More-over the functional flow of this application is derived graphically as well as detailed screenshots are provided for better understanding.

Assignment Report

Micro-kernel architecture using OSGi framework

Table of Contents

[1. Scenario 1](#_Toc4830744)

[i. Sports and other facilities handling 1](#_Toc4830745)

[ Basic flow of the app function 1](#_Toc4830746)

[ Detailed flow description 1](#_Toc4830747)

[ii. Reception Handling 1](#_Toc4830748)

[ Basic flow of the app function 1](#_Toc4830749)

[ Detailed flow description 1](#_Toc4830750)

[iii. Car-Hire service handling 1](#_Toc4830751)

[ Basic flow of the app function 1](#_Toc4830752)

[ Detailed flow description 1](#_Toc4830753)

[iv. Food and Beverages handling 1](#_Toc4830754)

[ Basic flow of the app function 1](#_Toc4830755)

[ Detailed flow description 1](#_Toc4830756)

[2. Manifest Implementation 1](#_Toc4830757)

[i. Sports and other facilities handling 1](#_Toc4830758)

[ii. Rooms handling(Reception service) 1](#_Toc4830759)

[iii. Car-Hire service handling 1](#_Toc4830760)

[iv. Food and Beverages handling 1](#_Toc4830761)

[3. Screenshots 1](#_Toc4830762)

[i. Sports and other facilities handling 1](#_Toc4830763)

[ii. Reception handling 1](#_Toc4830764)

[iii. Car-Hire service handling 1](#_Toc4830765)

[iv. Food and Beverages handling 1](#_Toc4830766)

[4. Project implementation procedure 1](#_Toc4830767)

# Scenario

(Figure 1: Graphical view of the system)

This Hotel system is divided into four main sub sections as above. These sections will work in producer-subscriber bundles to do calculations, manage data and provide outputs to the user.

## Sports and other facilities handling

### Basic flow of the app function

(Figure 2 : Basic flow graph)

### Detailed flow description

1. After starting the program, Main menu which consists sub systems will be displayed (Figure 13 : Hotel Homepage)
2. After clicking the ‘Facilities’ button, an input box will be prompted the user asking about the specific facility he/she wants.(Figure 14 : Sub-services offered)
3. Upon selecting a valid facility option, user will be asked to enter several inputs as their requirement.(Figure 15 : Consuming time period prompt)
4. When user hits enter from step 3, A dialog box will be prompted containing the total amount that should be paid.(Figure 16 : Payable amount prompt)

## Reception Handling

### Basic flow of the app function

### Detailed flow description

Reception menu has 2 options (Figure 17: Reception Page):

* View Rooms
* Book Room

## 

|  |  |
| --- | --- |
| Use case Id: UC01 | |
| Use case name: Book room | |
| Pre-Condition: CustomerConsumer,ReceptionProducer and ManagerProducer should be started | |
| Post Condition: Book receipt is generated | |
| Primary Actor: Customer | |
| Main Scenario | 1. Customer selects option 2 “Book room” from Reception menu. 2. System switches to Book room interface(Figure 19) 3. System asks to input customer personal details: name, email and contact no. 4. Further system asks to input category of room, check in and checkout time. 5. System asks to input details: number of adults, number of children and number of rooms. 6. System confirms booking. 7. System stores customer details. 8. System calculates room payment as well as generate receipt for customer. |
| Extensions | 1a) If customer does not type options of values other than 1 or 2, System displays “Please type valid option”.  6a) If name is not typed, System displays “Please enter name” and “Try again”. System does not continue after step 6.  6b) If customer does not type email, System displays “Please enter email” and “Try again”.  6c) If email is in invalid format where email does not contain @ and .com, System displays “Please enter valid email” and “Try again”. System does not continue after step 6.  6d) If contact number is not typed by customer, System displays “Please enter contact number” and “Try again”. System does not continue after step 6.  6e) If contact number is in invalid format where contact number should start with 10 digits and starts with 0, System displays “Please type valid contact number” and “Try again”. System does not continue after step 6. |

|  |  |
| --- | --- |
| Use case Id: UC02 | |
| Use case name: View Rooms | |
| Pre-Condition: CustomerConsumer,ReceptionProducer and ManagerProducer should be started | |
| Primary Actor: Customer | |
| Main Scenario | 1. Customer types option 1 “View Rooms” from Reception menu. 2. System switches to rooms interface which displays category of room details: name, cost and description (Figure 18). |
| Extension | 1a) If customer does not type options of values other than 1 or 2, System displays “Please type valid option”. |

## Car-Hire service handling

### Basic flow of the app function

### Detailed flow description

1. The Car-hire main menu display 2 options [1.ViewCars || 2. BookCar]
2. ViewCars option display the available cars and payment for one day.
3. BookCar option asked some User inputs for rent a car.
4. At the end display the payment details.

## Food and Beverages handling

### Basic flow of the app function

### Detailed flow description

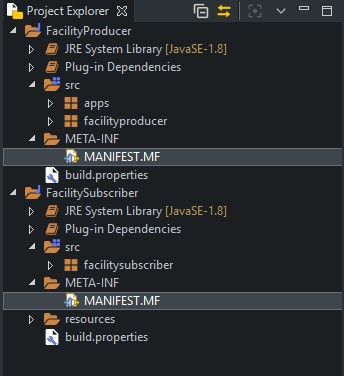
1. The meals menu will display 2 options (Figure 24: Meals home page).

* View Meals
* Order Meals

1. View Meals option will display the available meals with the price of 1 portion (Figure 24: View Meals).
2. Order Meals option will display the Meal numbers for each meal and ask the user to enter the desired number. After entering the Meal number the service will also ask the user to enter the amount of meals needed (Figure 25: Order Meals).
3. The final price will be displayed after step 3.

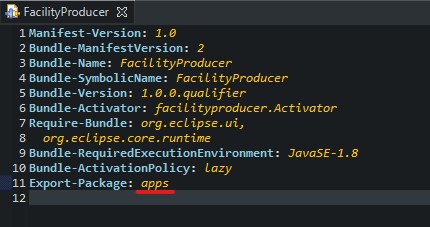
# Manifest Implementation

## Sports and other facilities handling



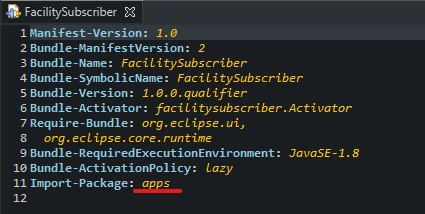
(Figure 3 : Project hierarchy)

* Producer/Publisher Manifest (Metadata)
  + Exported package: apps



(Figure 4 : Service publishers' 'MANIFEST.MF')

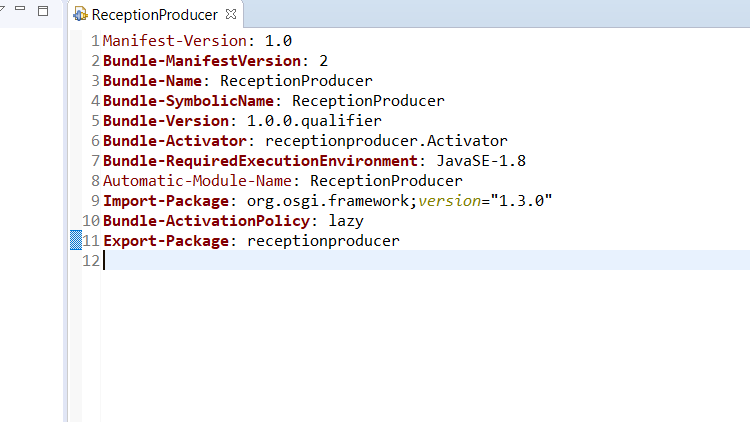
* Subscriber Manifest (Metadata)
  + Imported package: apps



(Figure 5 : Service subscribers' 'MANIFEST.MF')

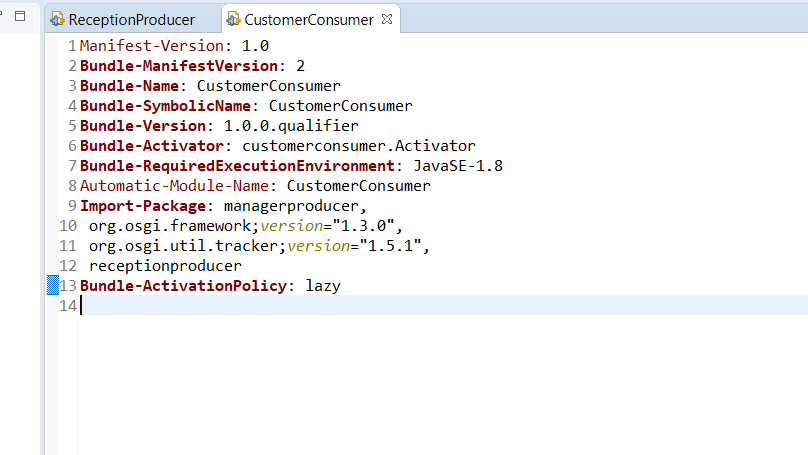
## Rooms handling(Reception service)

* Reception producer Manifest

****

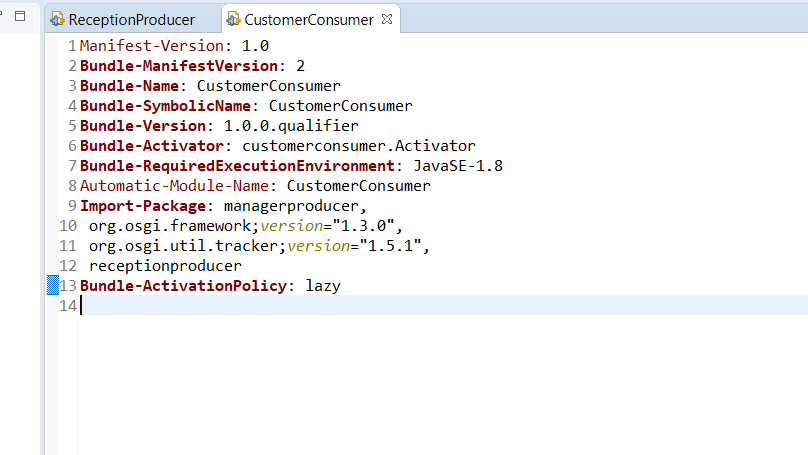
(Figure 6: Reception producers’ MANIFEST.MF)

* Customer consumer Manifest



(Figure 7: Customer consumers’ MANIFEST.MF)

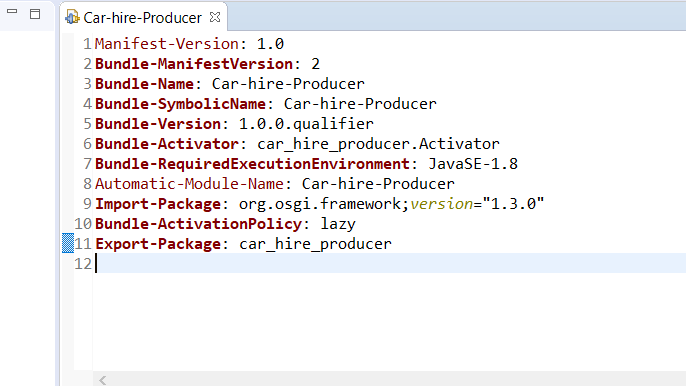
* Manager consumer - Manifest



(Figure 8: Manager consumers ‘MANIFEST.MF’)

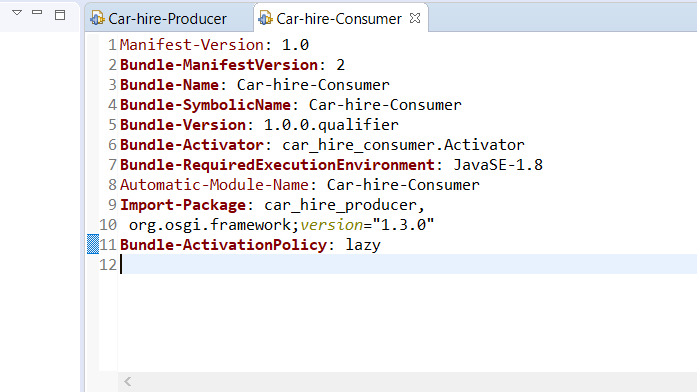
## Car-Hire service handling

* Car-Hire producer Manifest

****

(Figure 9: Car-Hire producer ‘MANIFEST.MF’)

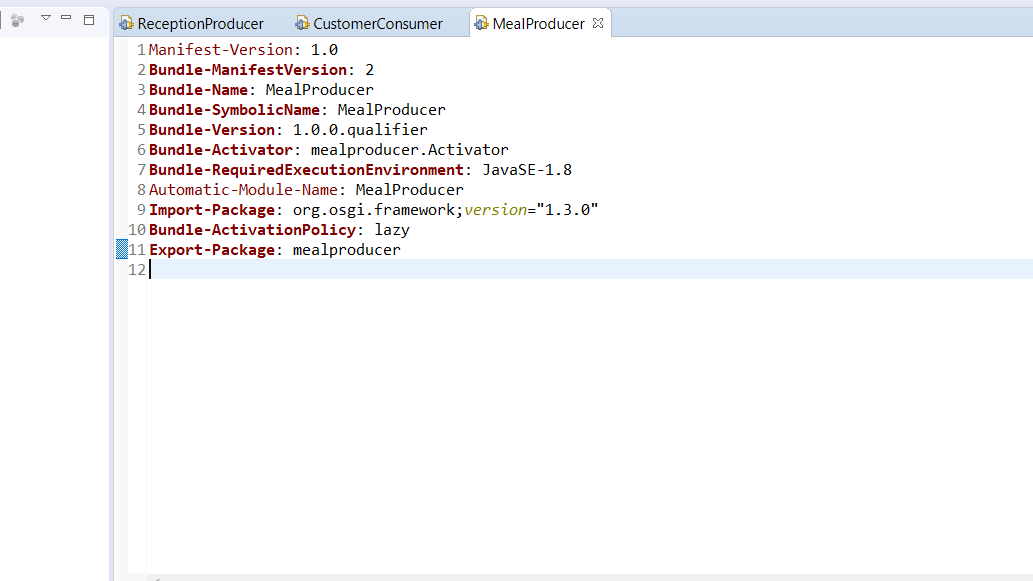
* Car-Hire consumer Manifest



(Figure 10: Car-Hire consumer ‘MANIFEST.MF’)

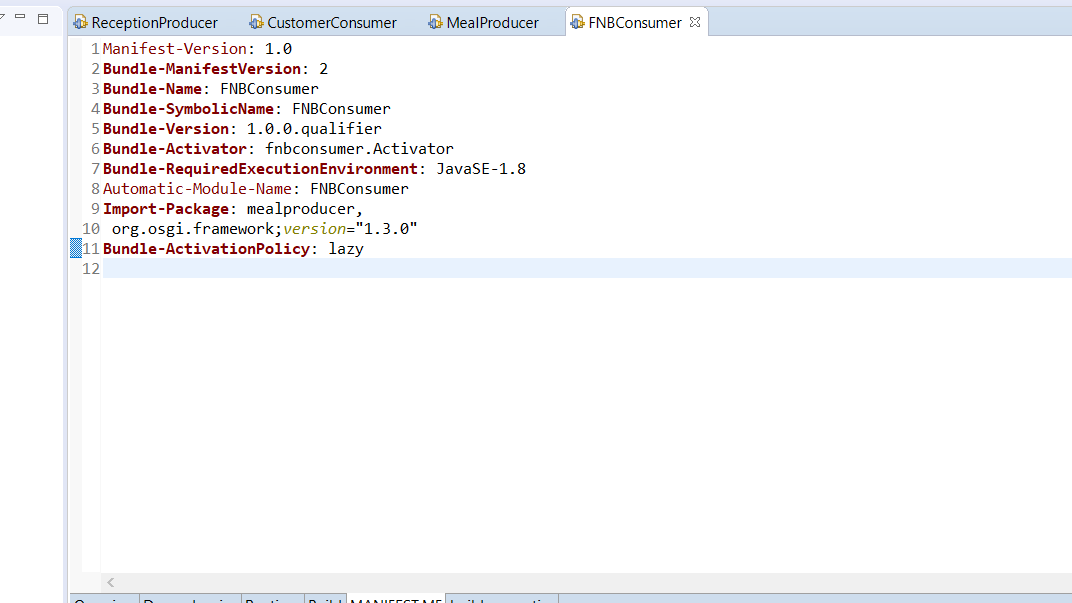
## Food and Beverages handling

* Meal service producer



(Figure 11: Meal service producers’ ‘MANIFEST.MF’)

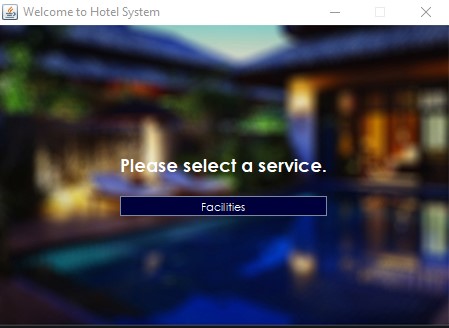
* Meal Service consumer



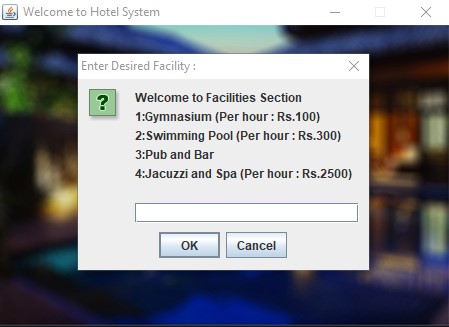
(Figure 12: Meal service consumers’ ‘MANIFEST.MF’)

# Screenshots

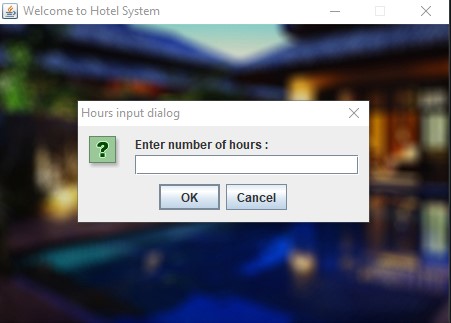
## Sports and other facilities handling



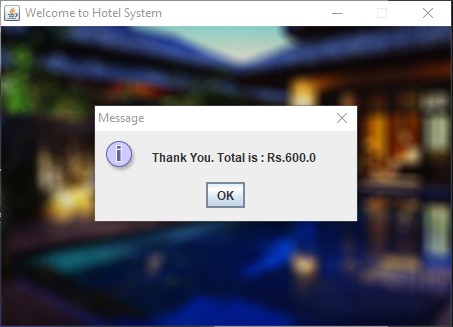
(Figure 13 : Hotel Homepage)



(Figure 14 : Sub-services offered)



(Figure 15 : Consuming time period prompt)

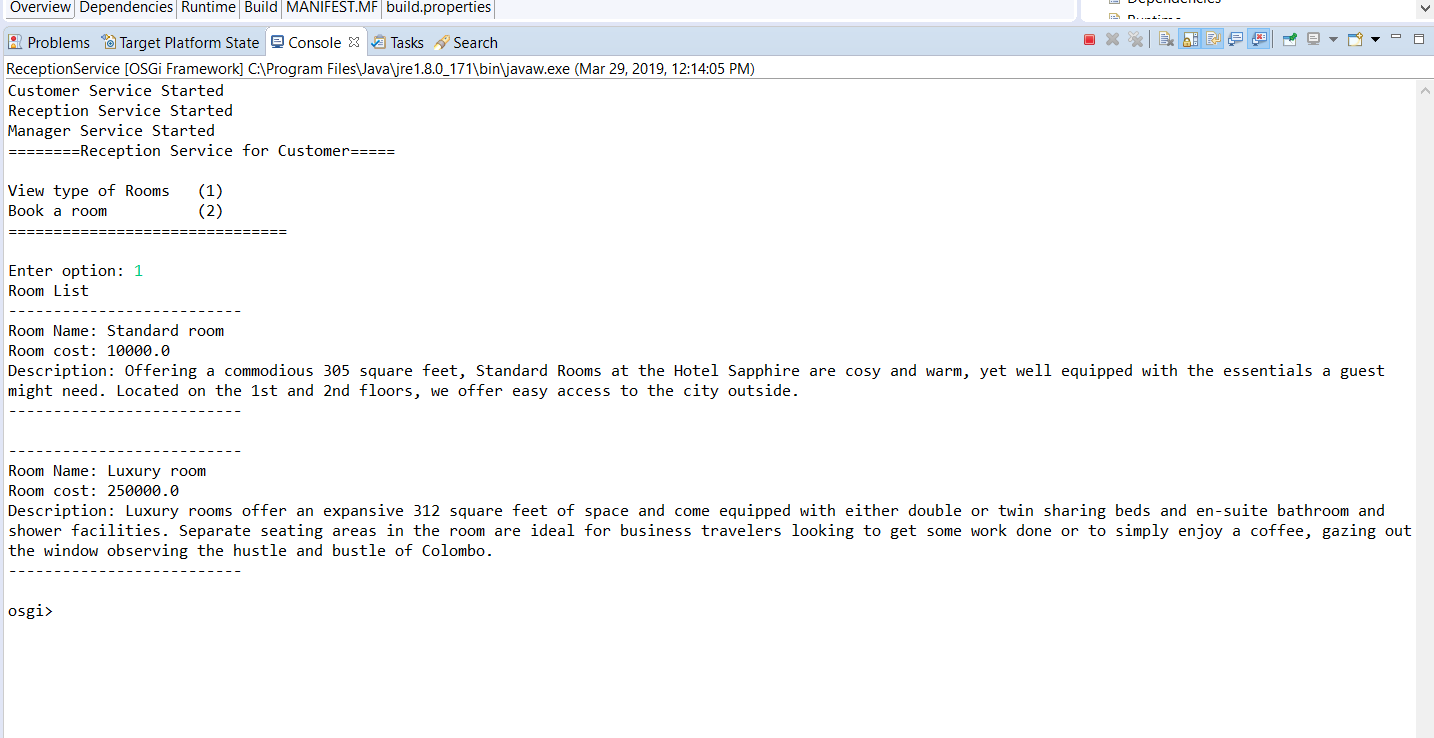


(Figure 16 : Payable amount prompt)

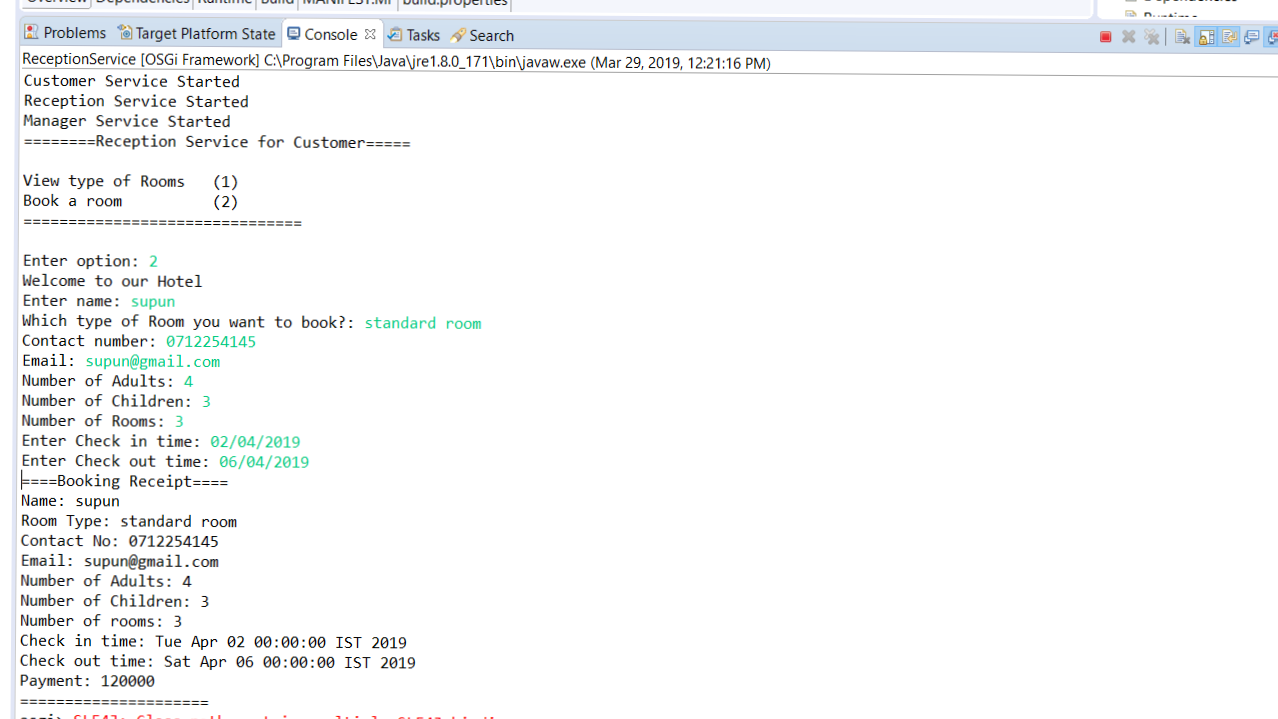
## Reception handling

****

(Figure 17: Start-up Menu)

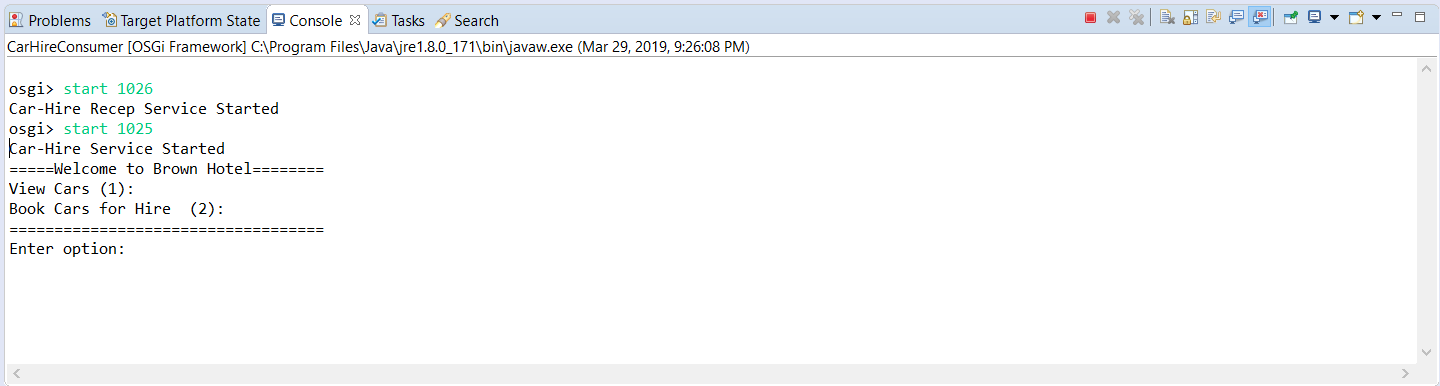


(Figure 18: View rooms)

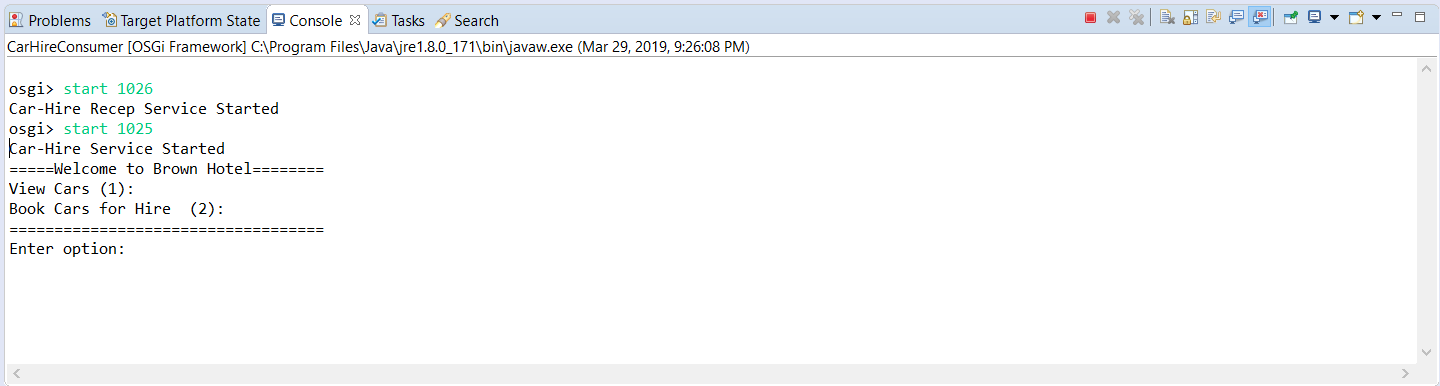


(Figure 19: Book room)

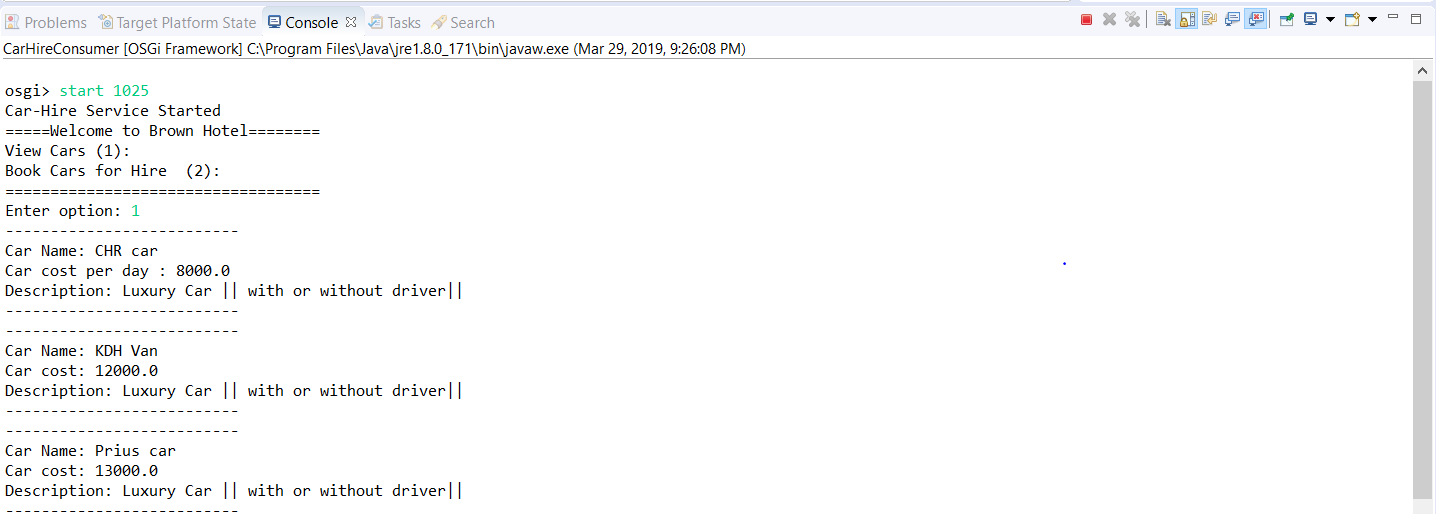
## Car-Hire service handling



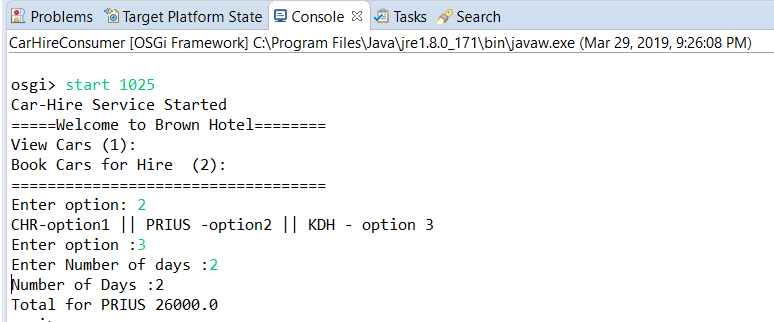
(Figure 20: Car-Hire service startup)



(Figure 21: Car-Hire service start-up)



(Figure 22: View cars)

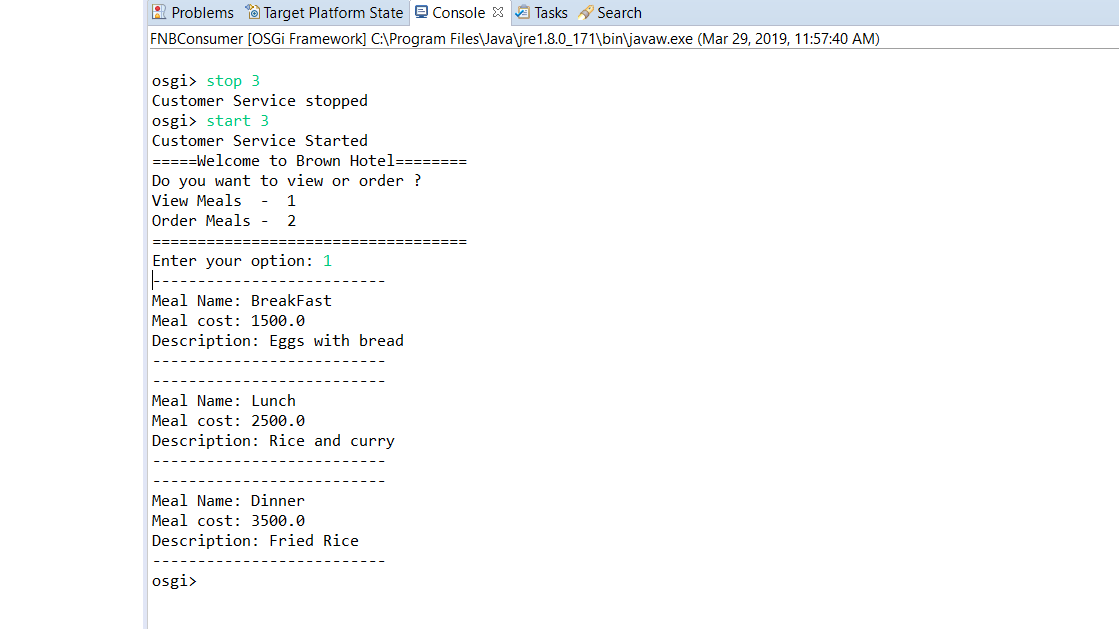


(Figure 23: Book cars)

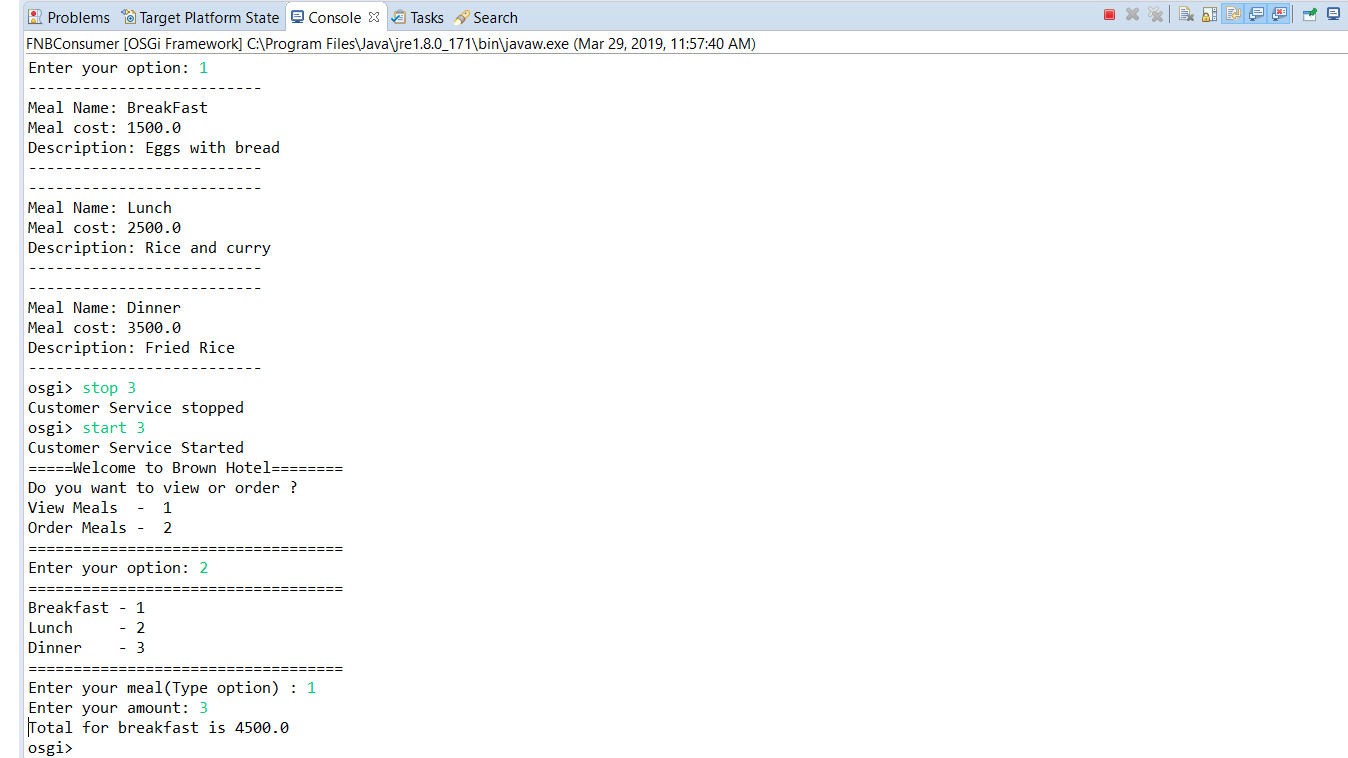
## Food and Beverages handling

****

(Figure 24: Meal service start-up)



(Figure 25: View meals)

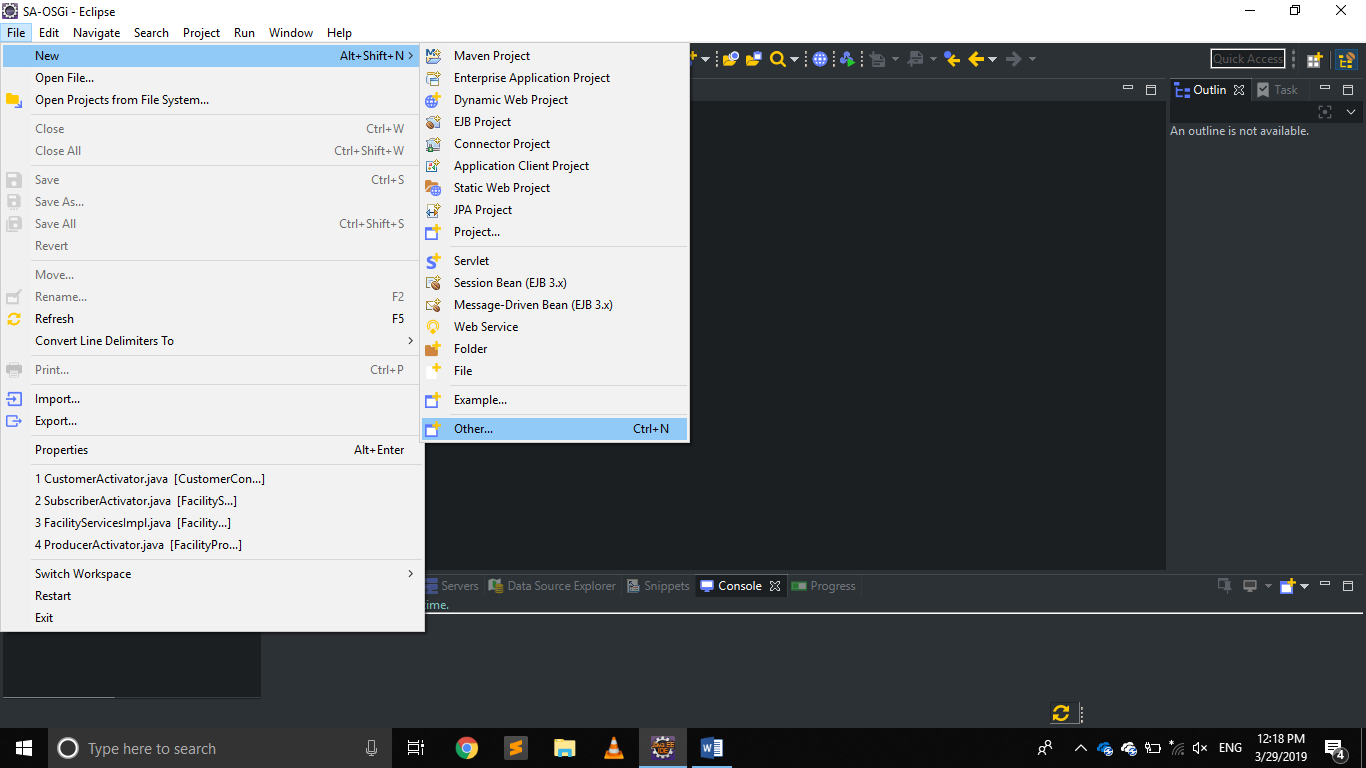


(Figure 26: Order Meals)

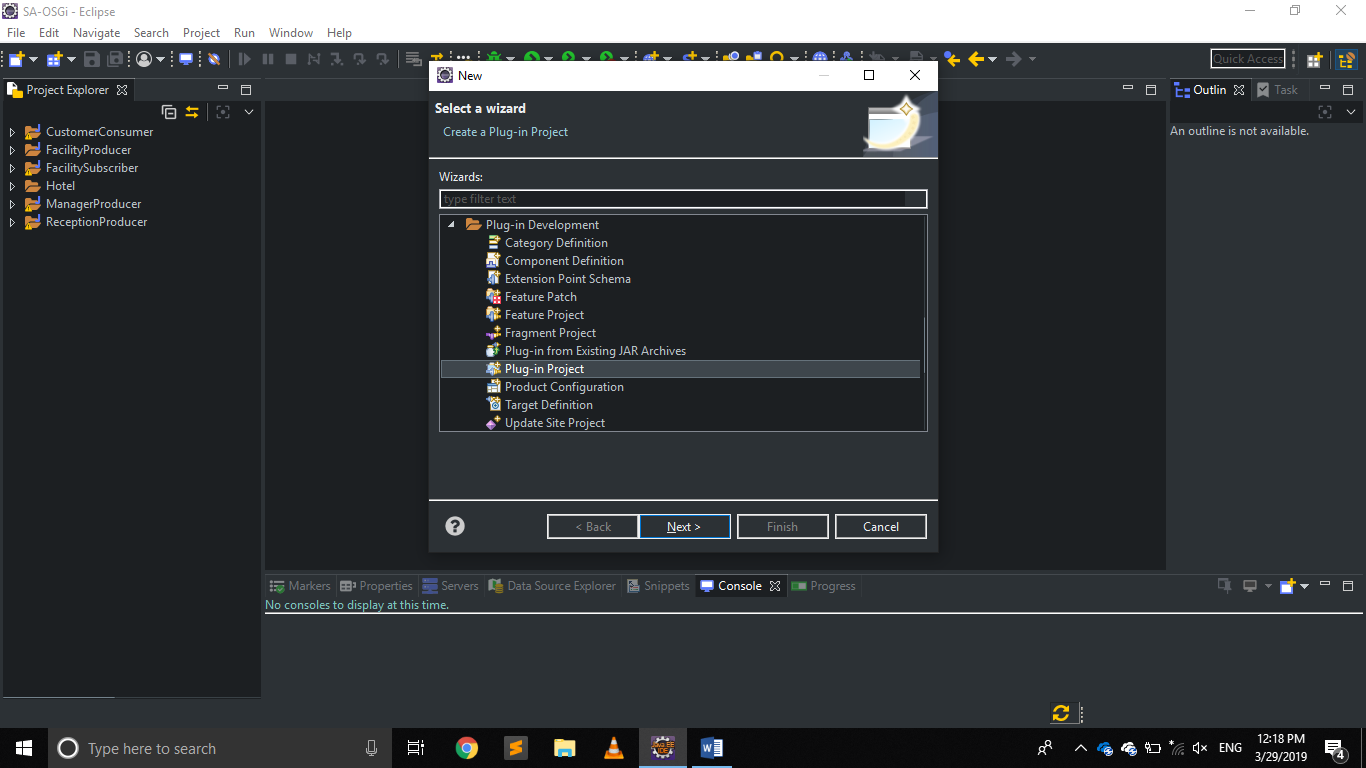
# Project implementation procedure

Producer Implementation

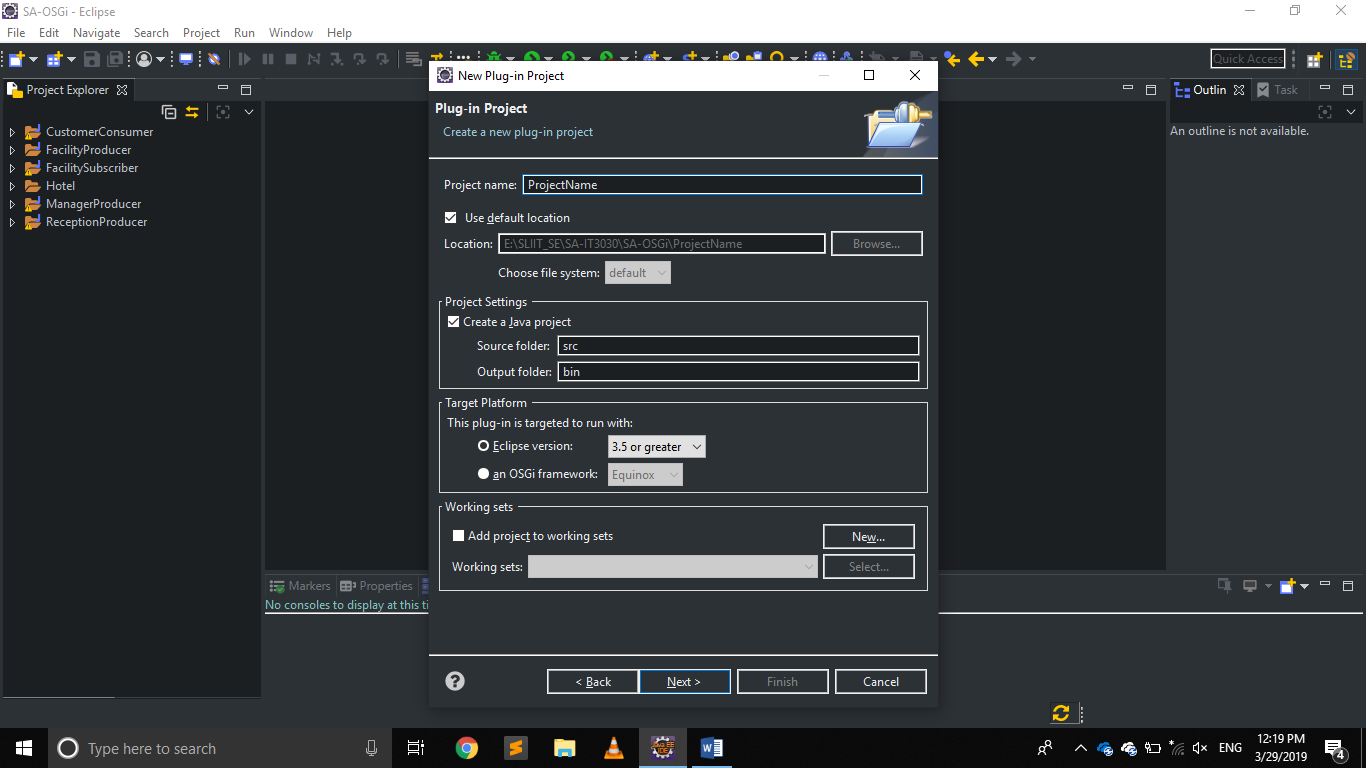
1. Create a ‘Plug-in Project’ from File > New > Other



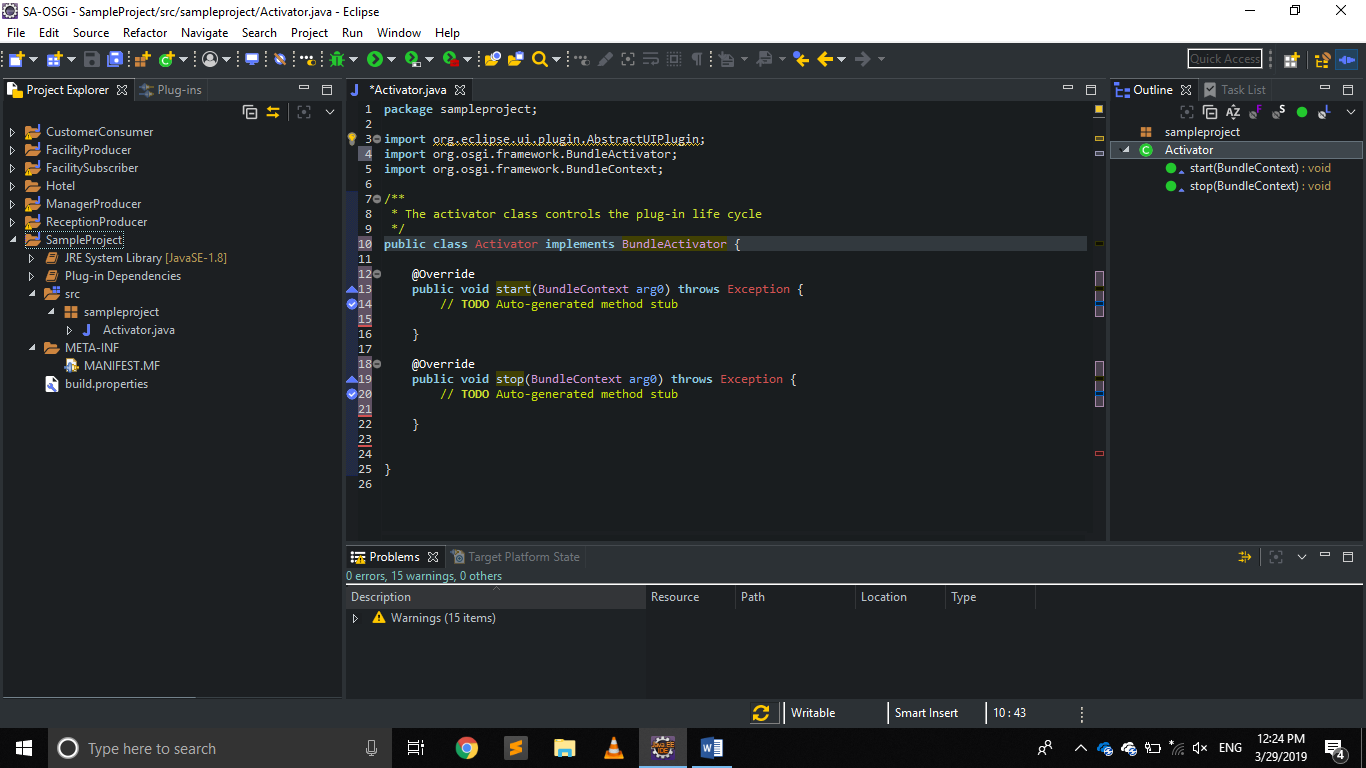
(Figure 27: Creating a new project)



(Figure 28: Selecting a 'Plug-in' project)



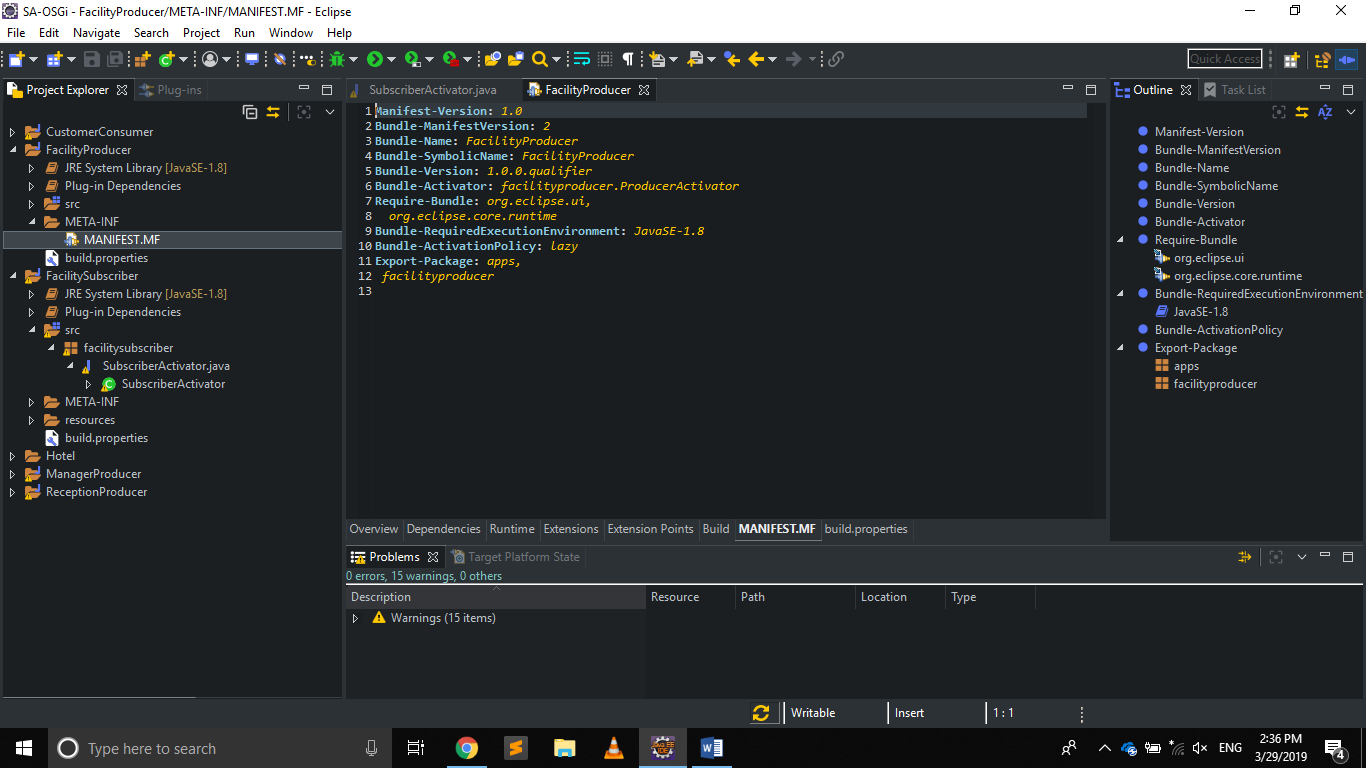
(Figure 29: Project details confirmation prompt)



(Figure 30: 'Activator' class of the created project)

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)

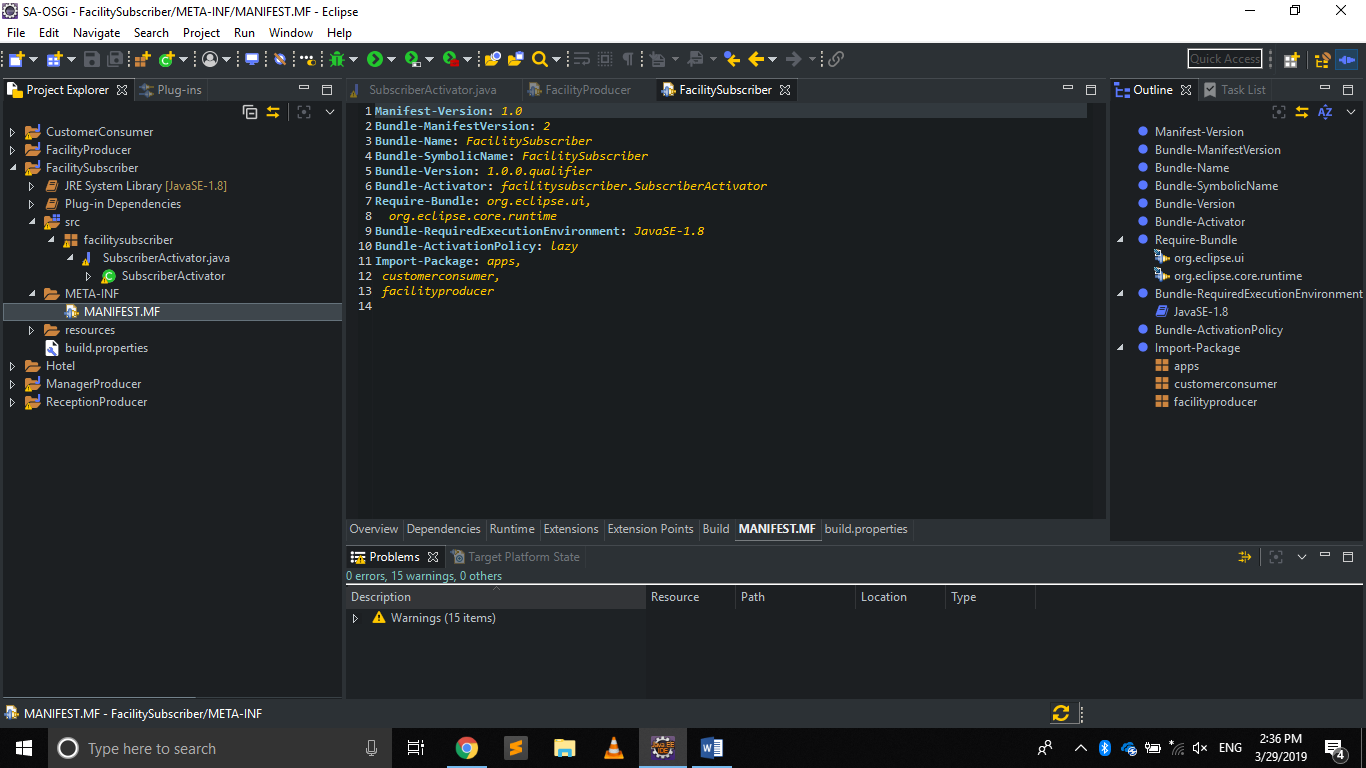


(Figure 31: Exporting packages from 'MANIFEST.MF')

Consumer Implementation

Repeat step 1 as in Producer to create a project.

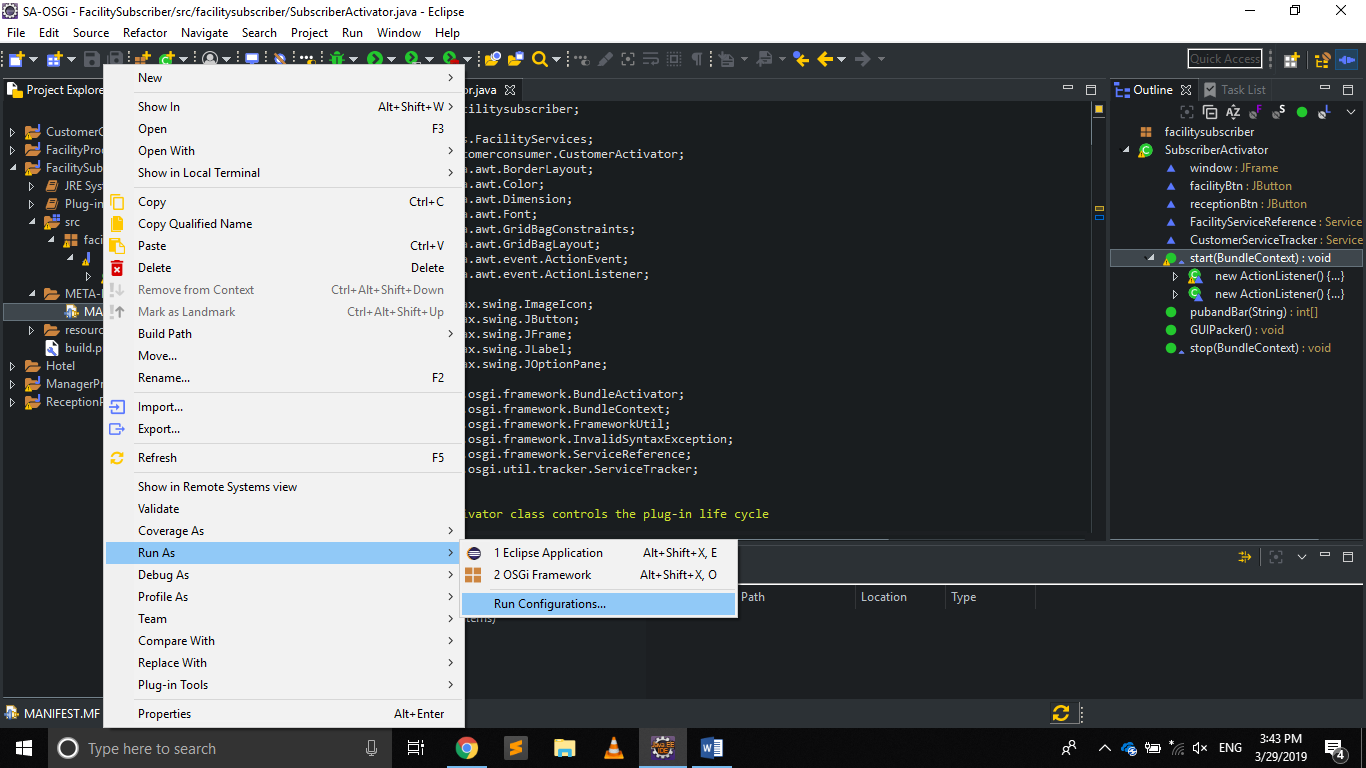
1. Import packages that have exported from Producers’ ‘MANIFEST.MF’ to the Consumers ‘MANIFEST.MF’



(Figure 32: Importing packages from Producer)

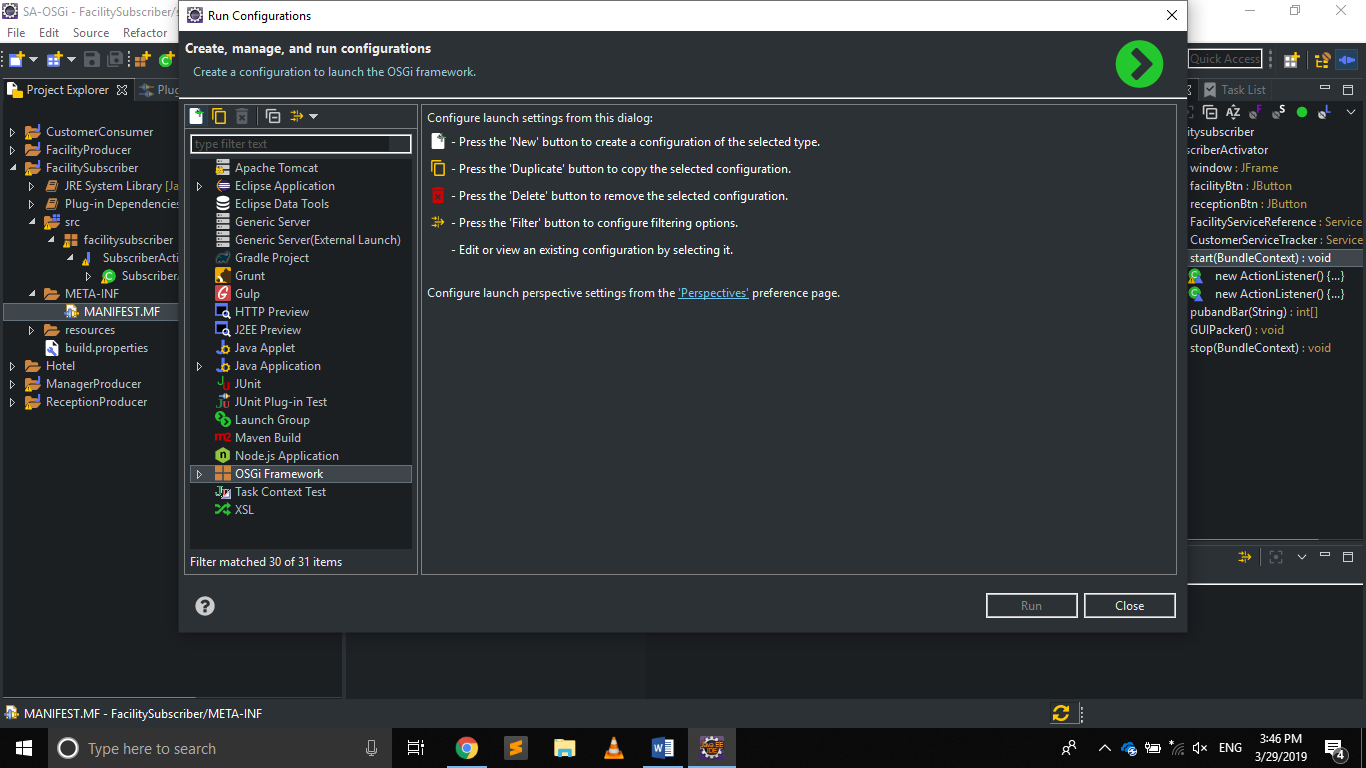
1. Create a separate run configuration to run Producer and Subscriber

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



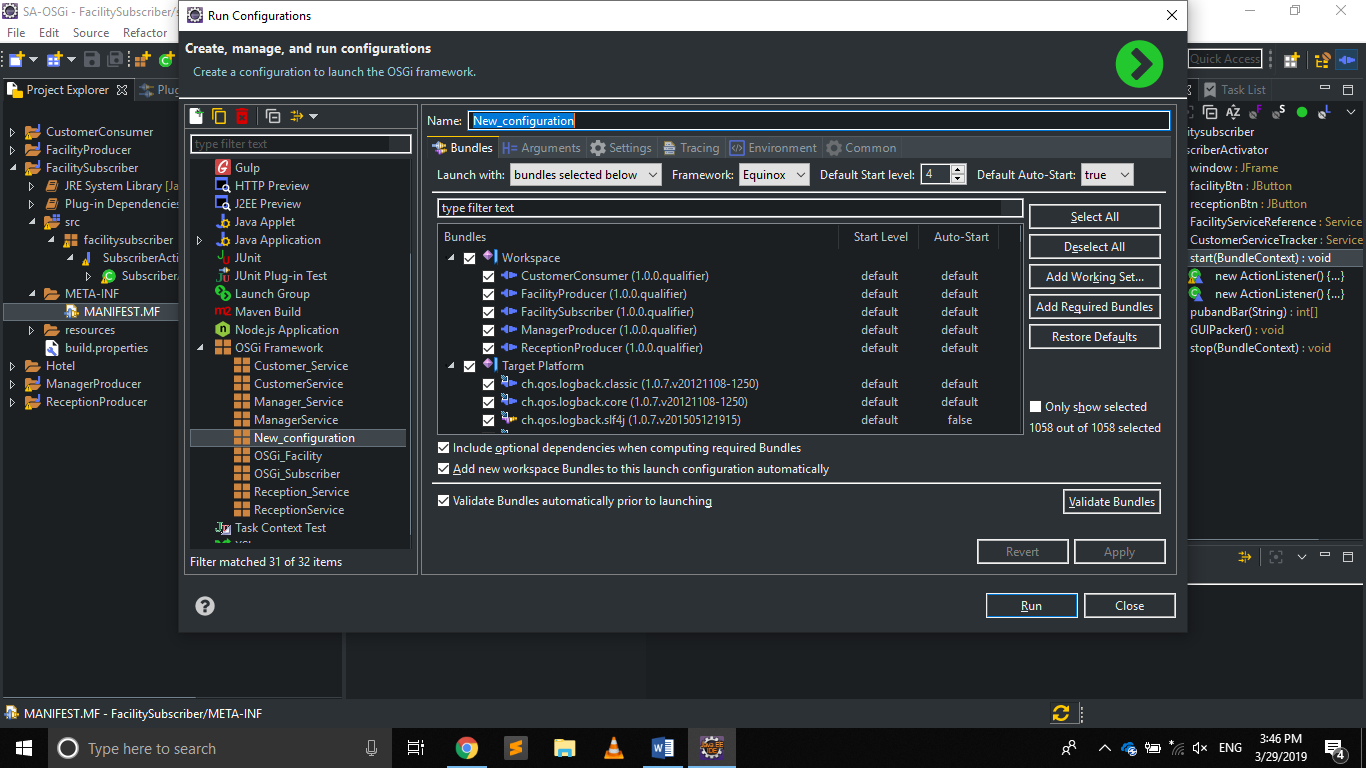
(Figure 33: Viewing run configurations)

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



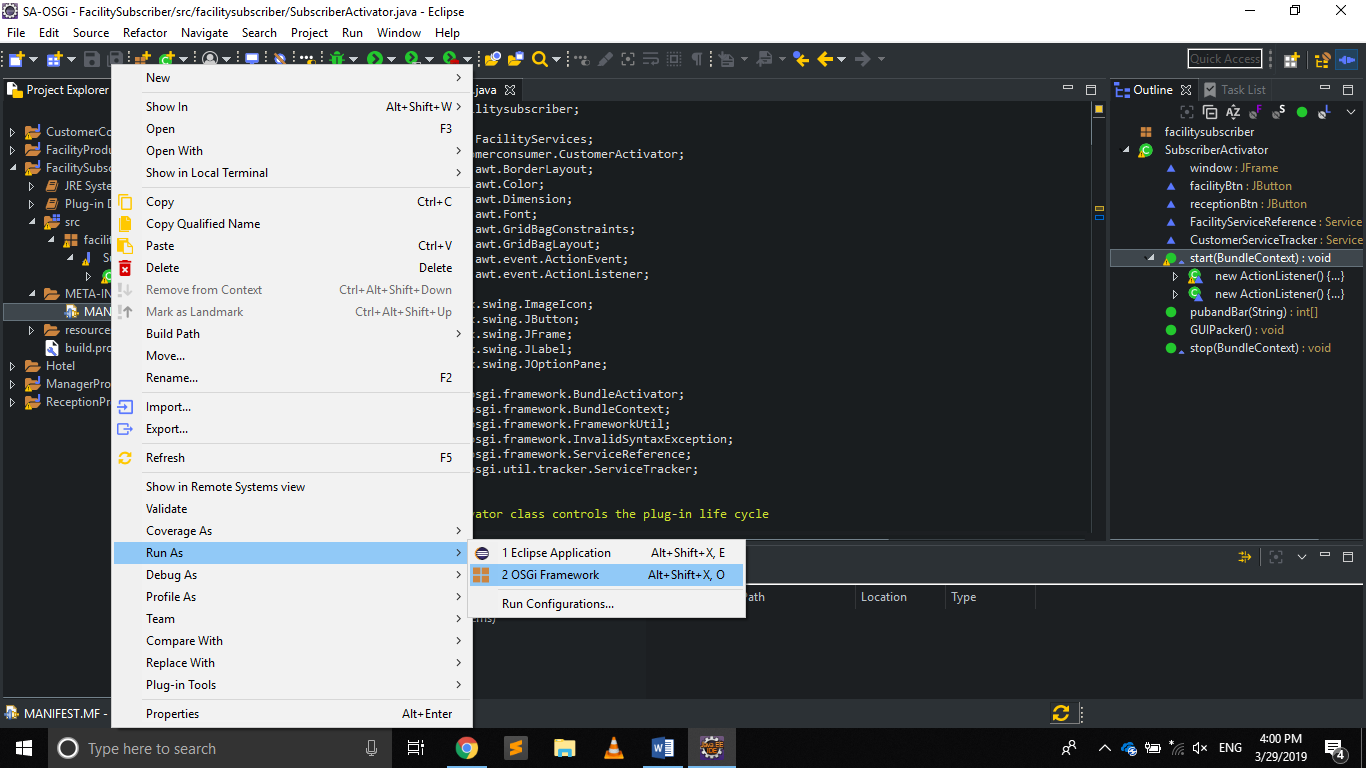
(Figure 34: Creating new OSGi configuration)

1. Add new name > Click ‘Run’



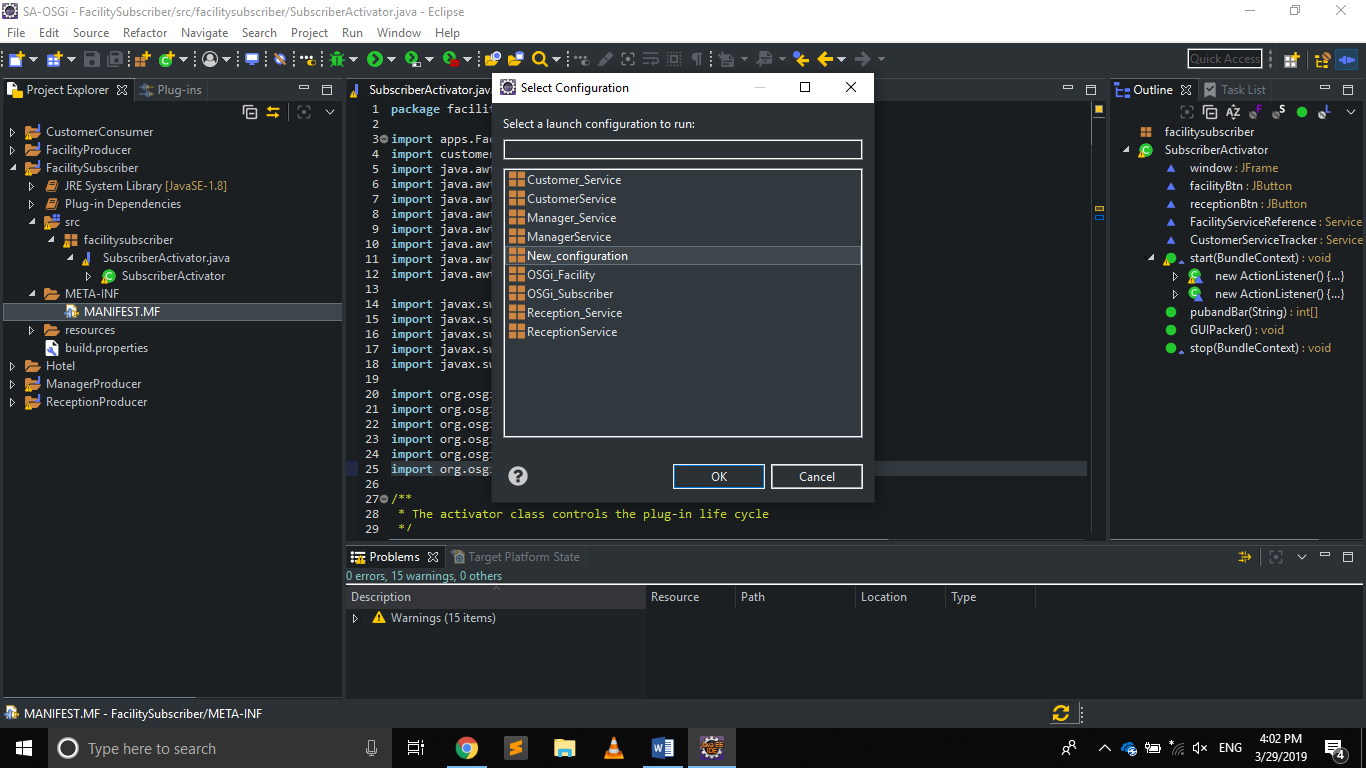
(Figure 35: Inserting a name to configuration)

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



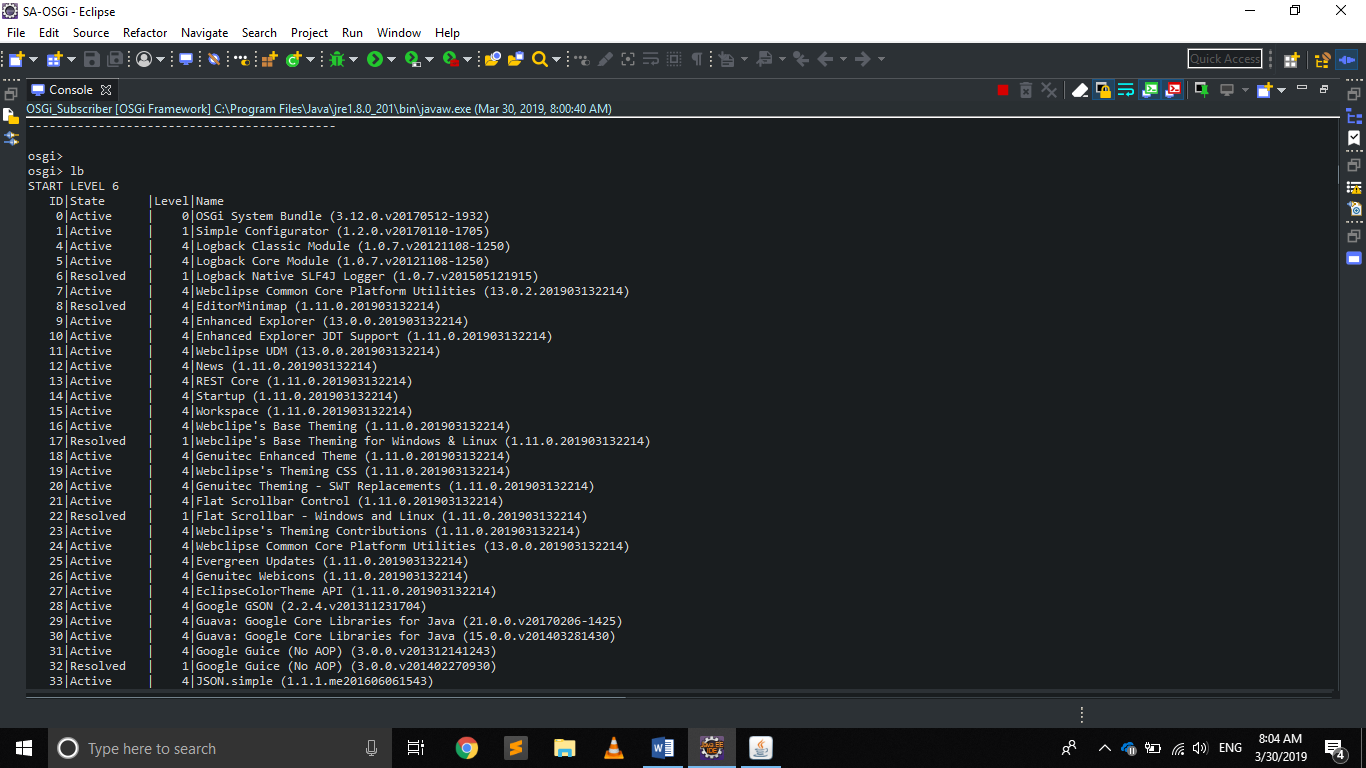
(Figure 36:Selecting run configuration)

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



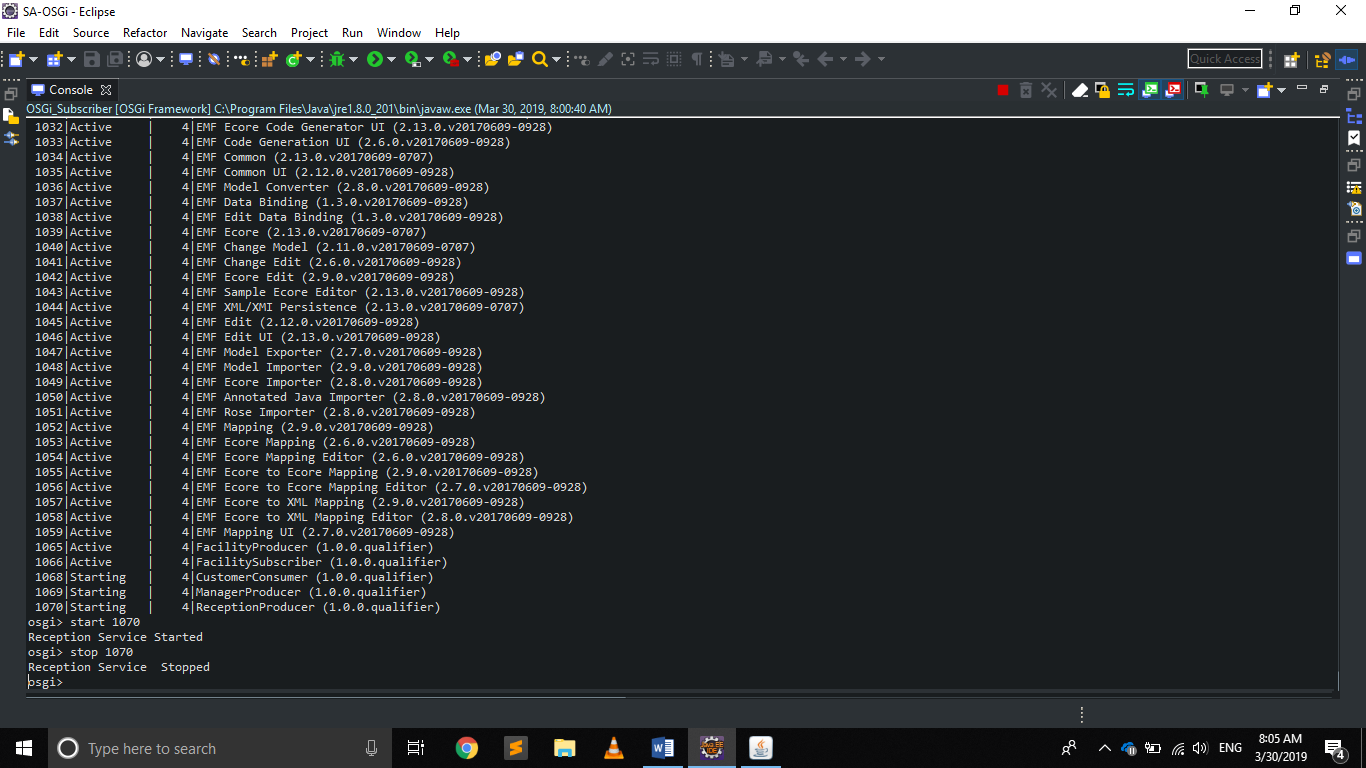
(Figure 37: Selecting run configuration from available OSGi configurations)

* 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)



(Figure 38: Issuing the ‘lb’ command to view/ list bundle details)

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



(Figure 39: Issuing the ‘start’ and ‘stop’ commands)