**COIT20259**

**Enterprise Computing Architecture**



**Prepared By:**

**Patel Gazal DineshBhai**

**12052661**

Contents

[**1.** **Design Document of Application** 3](#_Toc526256745)

[**1.** **Architecture of Application** 3](#_Toc526256746)

[**2.** **Description of tier component interaction and page navigation for Create a new car** 4](#_Toc526256747)

[**2.** **Test Instructions for Application** 5](#_Toc526256748)

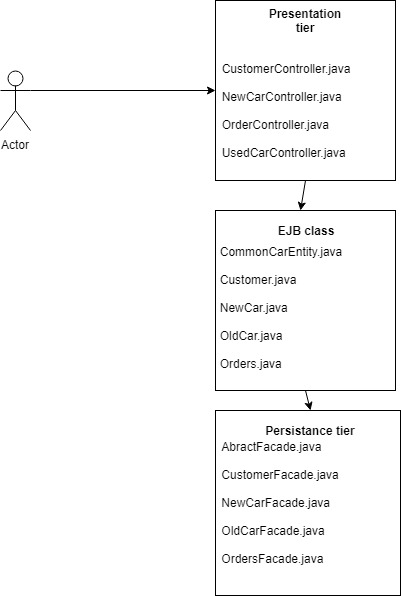
[**1.** **Instructions for compiling and deploying the project** 5](#_Toc526256749)

[**2.** **Inputs and expected outputs of each test** 6](#_Toc526256750)

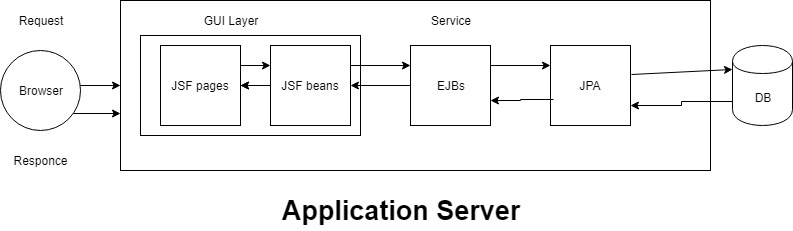
[**3.** **Coverage of the application scenario** 15](#_Toc526256751)

# **Design Document of Application**

## **Architecture of Application**

****

**Fig1: 3 Tiers architecture**

****

## **Description of tier component interaction and page navigation for Create a new car**

* When user click on Create a new brand car link in the index page it will navigate to the JSF page “createnewcar.xhtml”
* After entering appropriate information in createnewcar page, when user click on the button “Create a brand new car” it will call insert() method in NewCarController.java
* NewCarController call create() method in NewCarFaced and persist the data from JSF page into NewCar entity and NewCarController call findall() method and return the JSF page “listnewcar.xhtml”, which display the all stored new cars details

# **Test Instructions for application**

## **Instructions for compiling and deploying the project**

* To compile the project “CarBussiness”, user need to open the project in NetBeans and clean and build the project.
* To deploy project user need to follow the below steps:

Step 1: Open Command window and enter command startNetworkServer

Step 2: Open Command window and enter command asadmin start-domain

Step 3: After starting server user need to create connection pool.

To create connections pool user need to enter following command

asadmin create-jdbc-connection-pool

--datasourceclassname=org.apache.derby.jdbc.ClientDataSource

--restype=javax.sql.DataSource --property

portNumber=1527:password=APP:user=APP:serverName=localhost:dat

abaseName=WEBDB:connectionAttributes=create\=true

YOUR\_POOL\_NAME

Here in place of YOUR\_POOL\_NAME, user can enter any name to create pool

Step 4: After creating connection pool, to check whether Glassfish server is ready or not

Use the following statement

asadmin ping-connection-pool YOUR\_POOL\_NAME

Step 5: After that to create data source use the following statement

asadmin create-jdbc-resource --connectionpoolid YOUR\_POOL\_NAME

WEBDB

Step 6: Now to deploy project “CarBussiness” user need to open browser and type

The URL <http://localhost:4848>

Then click on the link “Application” in that page

Then Click on “Deploy” button

Then click on “Choose File” button and press “OK” button

(User need to select .war file from the folder “dist” in CarBussiness project

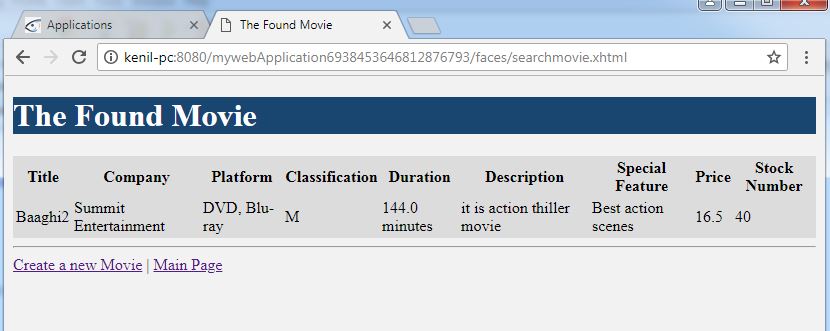
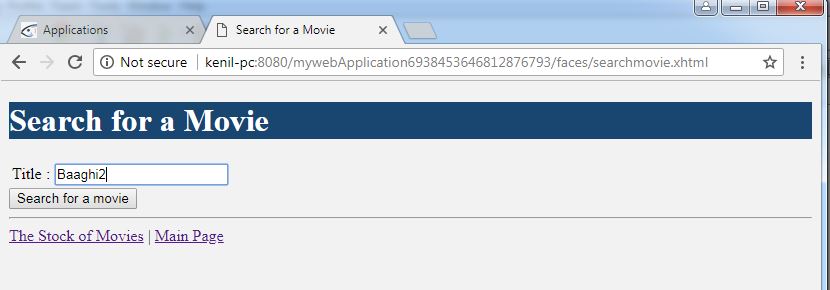
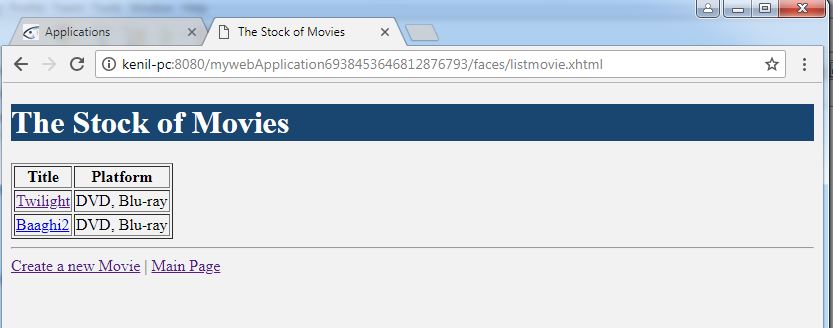
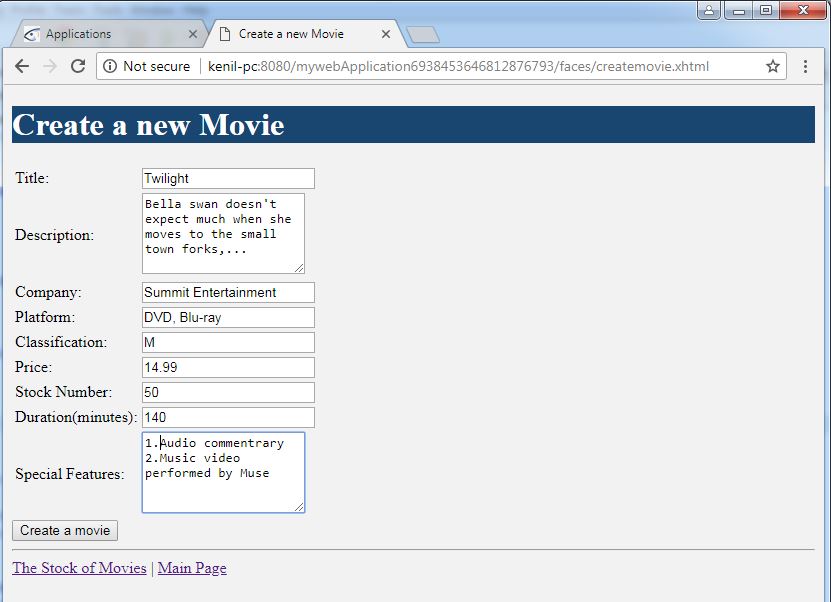
Folder)

Step 7: Then click on the “launch” link to start application.

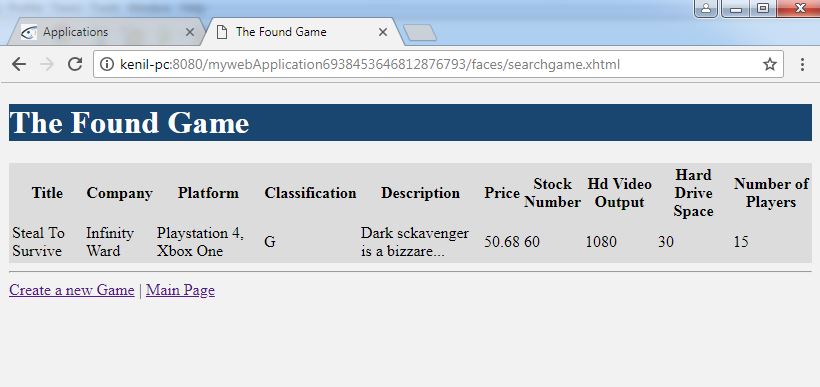
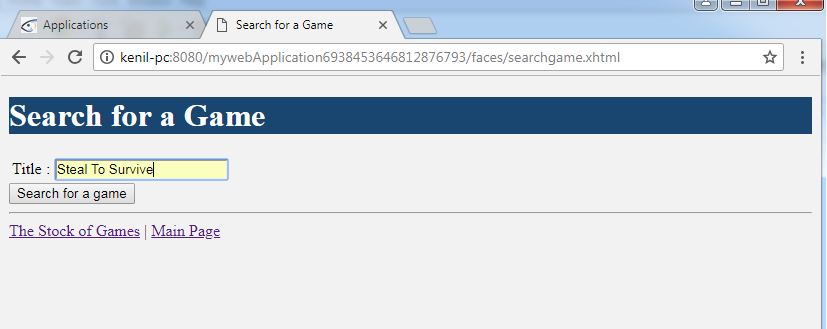
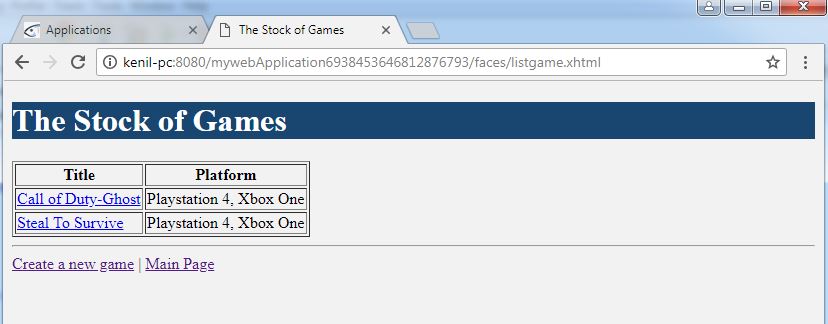
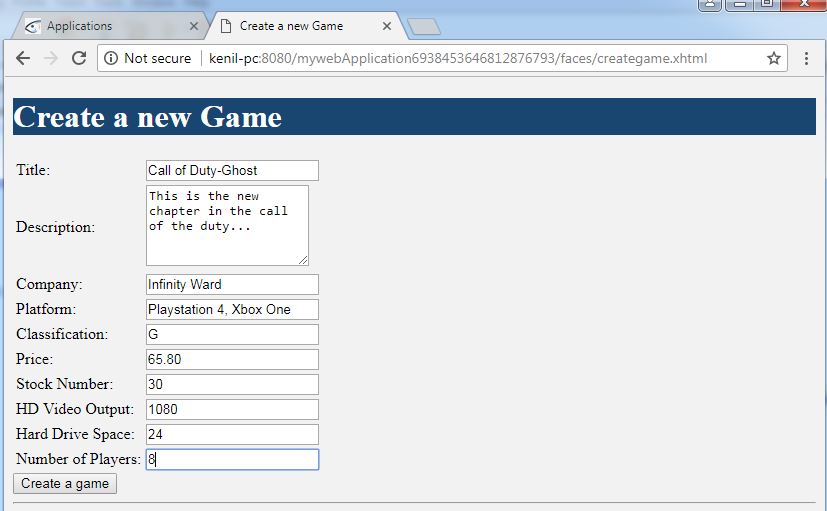
## **Inputs and expected outputs of each test**

Here below are the examples of sample inputs and expected results for that inputs.

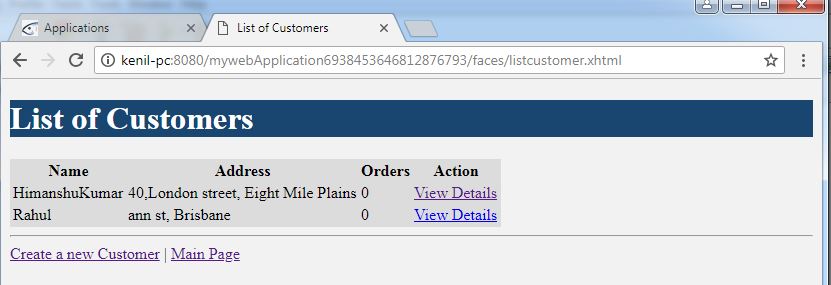
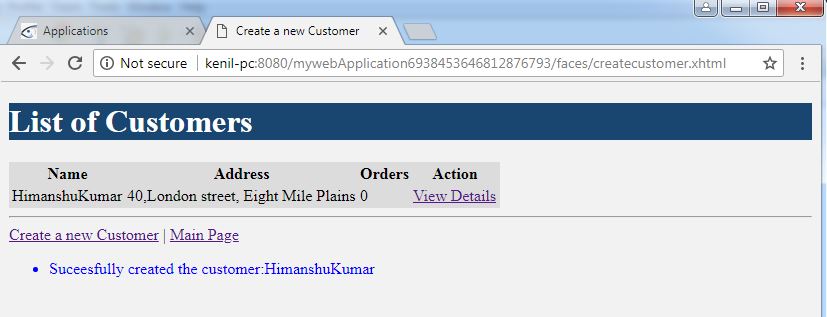
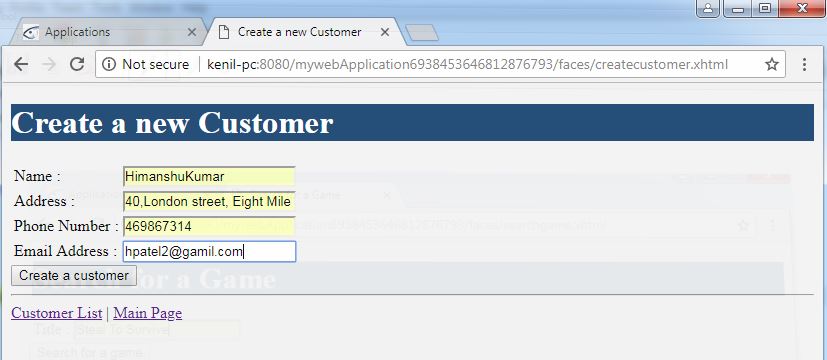
**Test for NEW Car**: Create, Stocks and Search



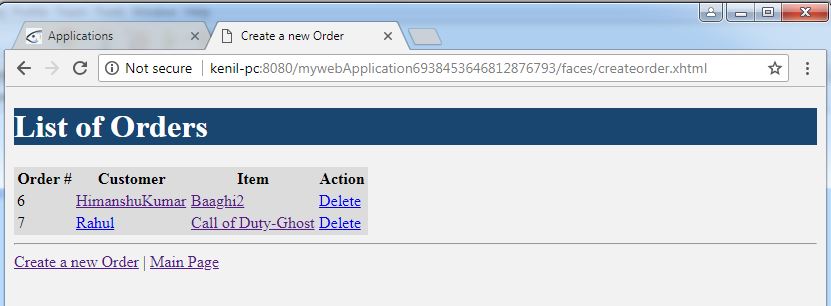
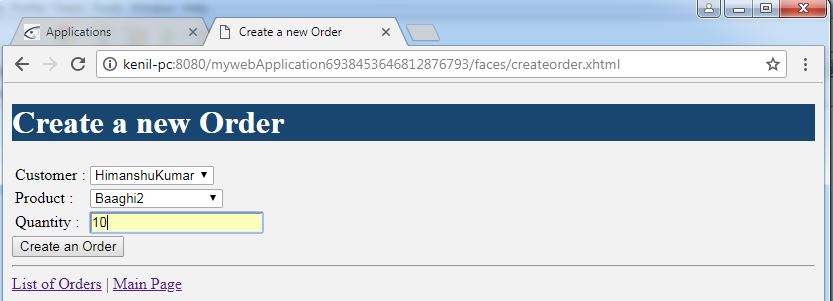
**Test for OLD car**: Create, Stocks and Search



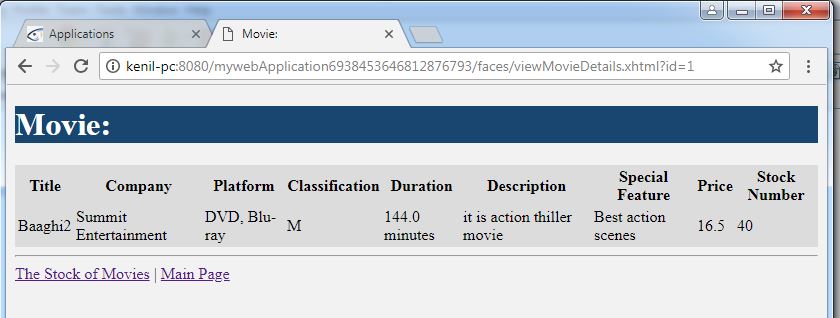
**Test for Customer**: Create, Stocks and Search



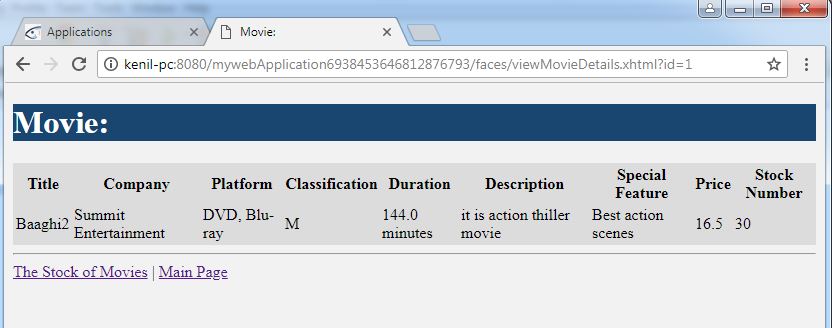
**Test for Order**: Create, Stocks and Search



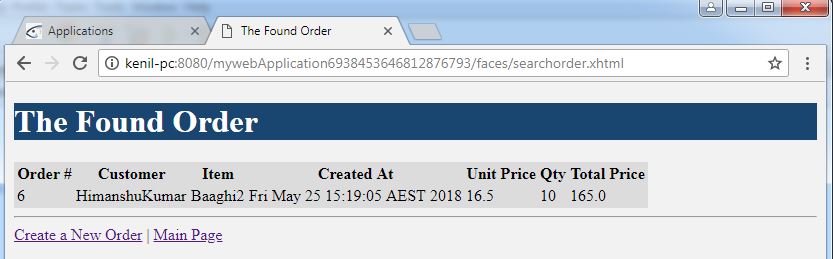
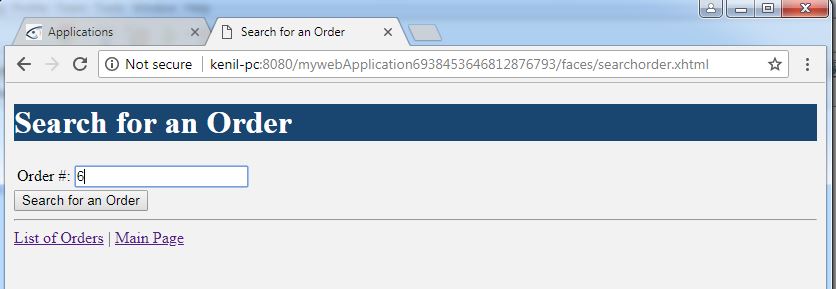
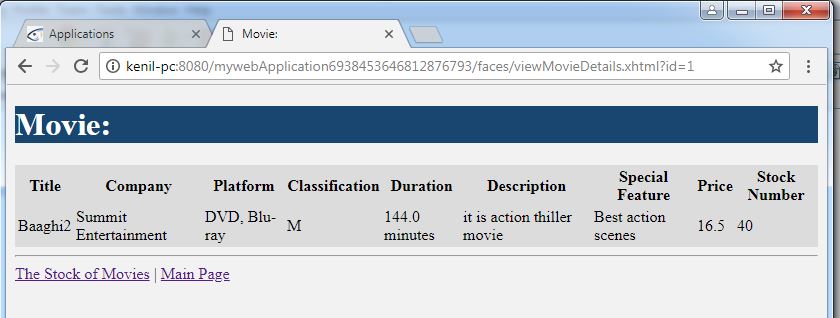
Car Stocks before Creating Order



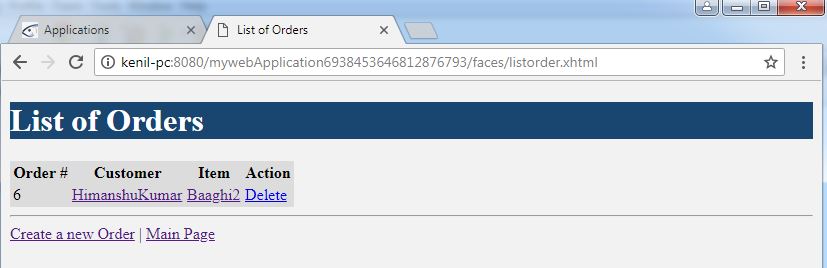
Car Stocks after Creating Order



Car Stocks after deleting Order



List of Orders after deleting order



## **Coverage of the application scenario**

This applications is the 3 tiers enterprise web applications. It successfully create New car, Old car, customer and order. It can all display the list of stored New car, Old car, customer and order from the database. User can search New car, Old car by title and customer by name and order by id.