Java Servlets 3.0
Lab Book

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Document Revision History

Date	Revision No.	Author	Summary of Changes
May 2015	4.0	Yukti A Valecha	Revamped from Servlets 2.5 to Servlets 3.0
May 2016	4.1	Yukti A Valecha /Anjulata	Revamped as per revised course contents

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Table of Contents

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Getting Started

Overview

This lab book is a guided tour for learning Servlets 3.0. It comprises scenario based applications and 'To Do' assignments. Flow diagrams and screen snap shots are provided where necessary.

Setup Checklist for Servlets 3.0

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP.
- Memory: 32MB of RAM (64MB or more recommended)
- Java SE version 8
- Internet Explorer 6.0 or higher
- Connectivity to Oracle database
- WildFly server Version 8.1.0

Please ensure that the following is done:

- Eclipse_JEE_Luna (4.x) is installed.

Instructions

- All lab assignments should refer coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory servlet_assgn. For each lab exercise create a directory as lab <lab number>.
  You may also look up the on-line help provided in JEE documentation

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Lab 1. Servlet Basics

Goals	• Understanding Servlet Basics and Invocation
Time	0.5 Hour

1.1 Create a Servlet program that will print the system date and time.

Note: The Servlet should be invoked in the following ways**:**

- On clicking of Hyper Link in html page
- On clicking of button in html page
- On typing the servlet URL pattern directly in Address bar of browser

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Lab 2. Request and Response Objects

Goals	• Understanding the request and response objects
Time	2 Hours

2.1: Design a login Page accepting username and password in HTML. The credentials need to be authenticated in Servlet.
The credentials could be validated by using hard coded values for username and password.
If user is valid then display HTML page as “Success” and display HTML page as “Failure” if user is invalid.

Note:

- Checking could be done by using private method inside servlet.
- The redirection to HTML pages is done using response.sendRedirect(String URLPattern) method

Refer below figure for HTML Login Page:

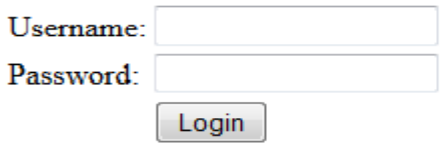

Figure 1

In extension to the previous assignment print the following details extracted from the ‘Request’ object….

1. MIME type accepted by the client
2. The locale setting
3. Data transferred in bytes

2.2: Consider a webpage which is displaying live stock market status. For such type of page, you would need to refresh your web page regularly using refresh or reload button with your browser.
Java Servlet makes this job easy by providing you a mechanism where you can make a webpage in such a way that it would refresh automatically after a given interval.
Consider the servlet code snippet below and complete the blank space left for automatic refresh to happen and auto-load page after 5 seconds.

Refer below figure:

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

```java
package com.cg.lab2.controller;
import java.io.IOException;
/**
 * Servlet implementation class Refresh
 */
@WebServlet("/Refresh")
public class Refresh extends HttpServlet {
   protected void doGet(HttpServletRequest request,
         HttpServletResponse response) throws ServletException, IOException {
      // TODO Auto-generated method stub
      // Set refresh, autoload time as 5 seconds

   }

}
```

Figure 2

Create a servlet that will redirect the current web application to ISPACE.

See below figure for same:

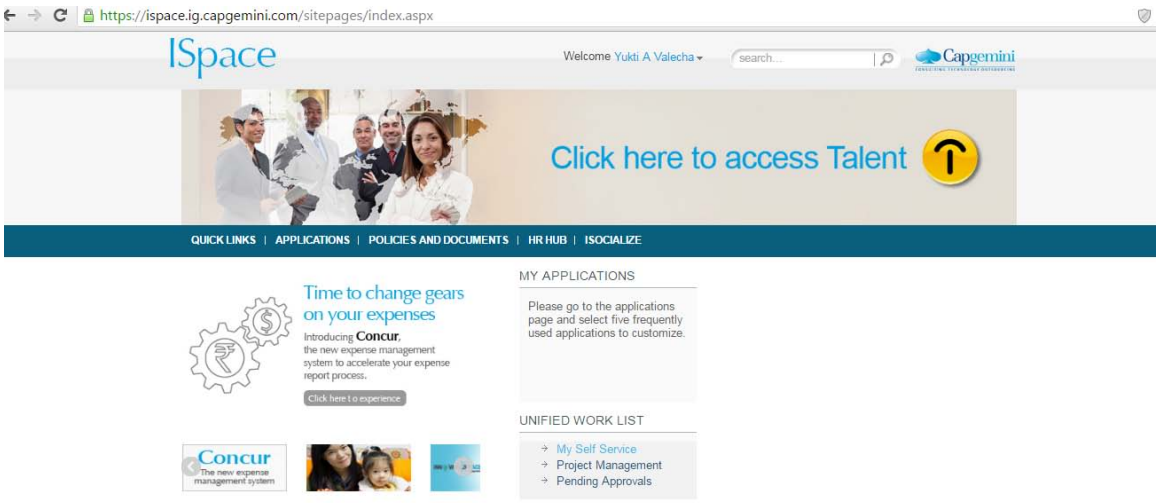

Figure 3

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Lab 3. Database Connectivity

Goals	• Creating DB connection pool with WildFly
Time	2 Hours

3.1: Steps to configure Database connection pool in WildFly:

Need to add Management User, Run D:\wildfly-8.1.0.Final\bin>add-user.bat
Create a Management User. Select Option (a)

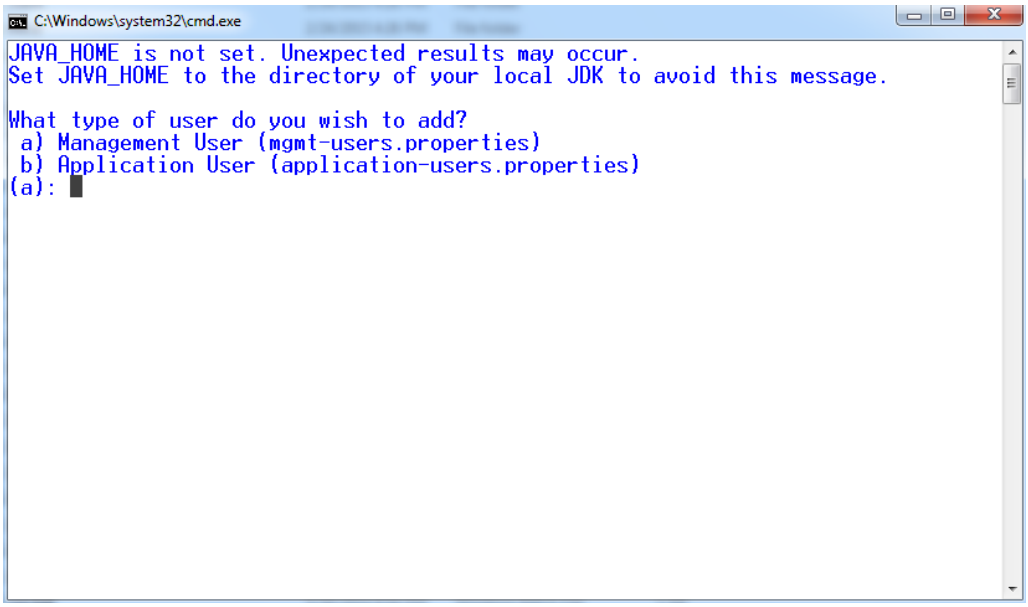

Figure 4

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Enter Username / Password

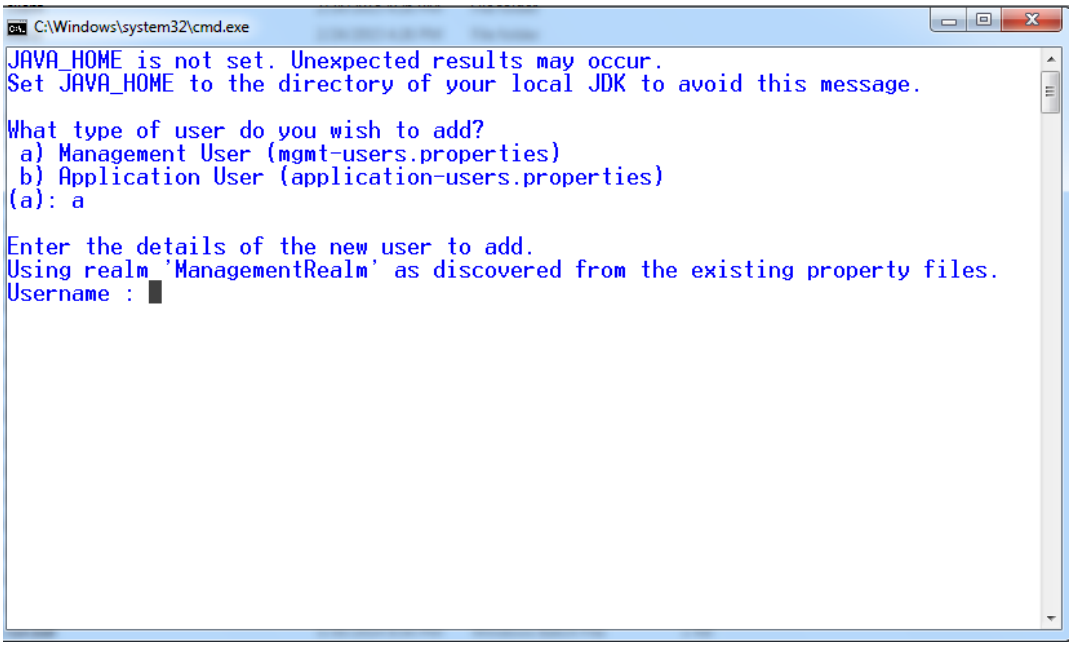

Figure 5

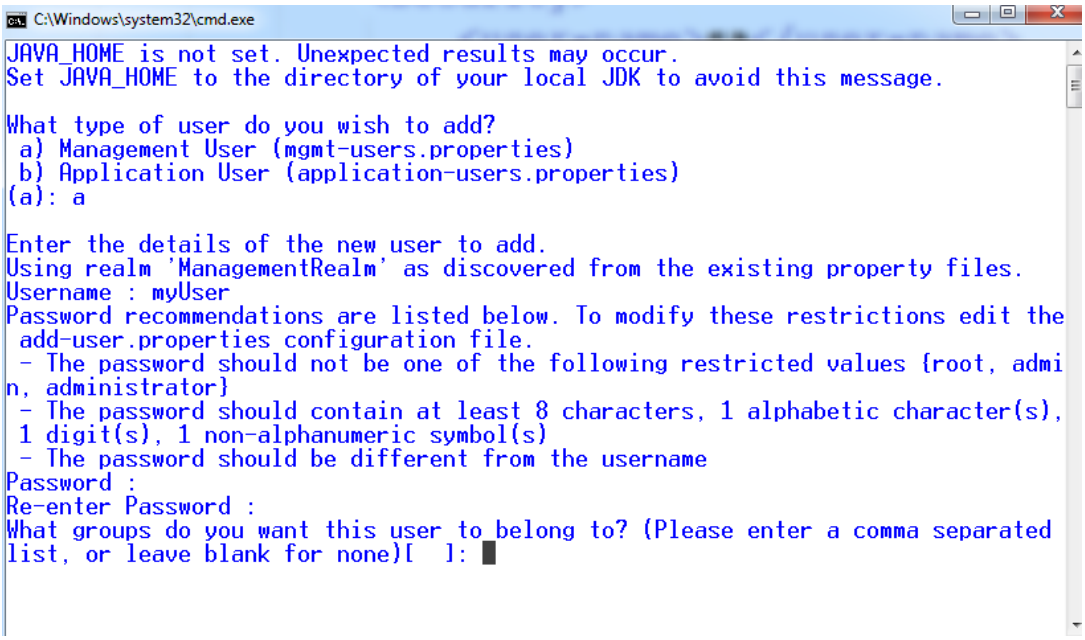

Figure 6

What Group this User should belong to – Leave as blank, Add the user to Management Realm with Option Yes.

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

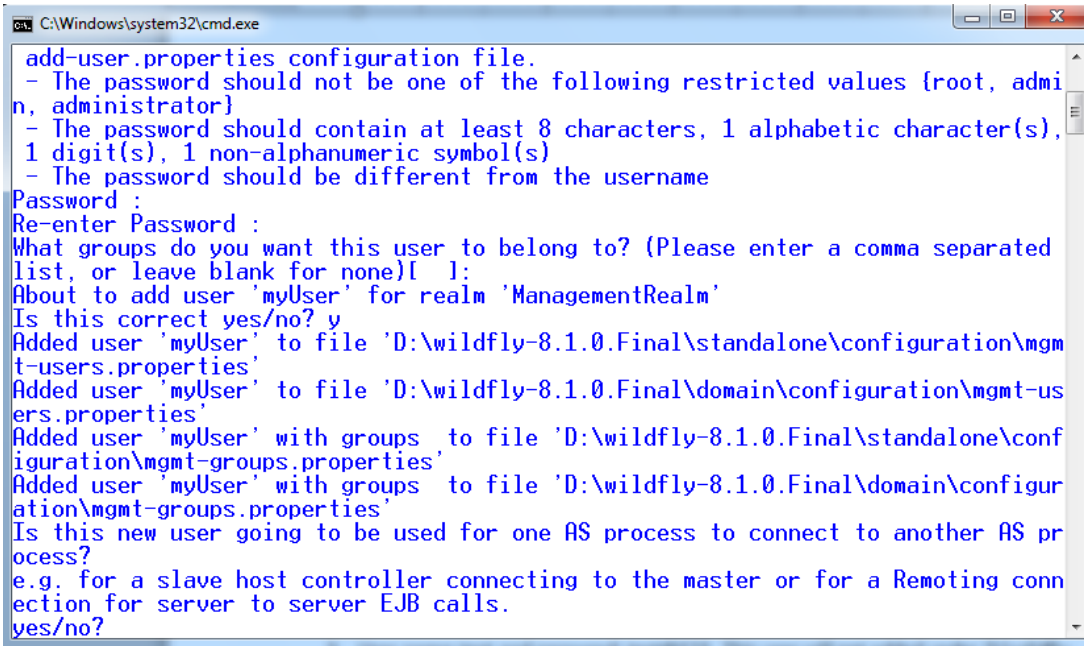

Figure 7

No need to add the User for server EJB Calls.

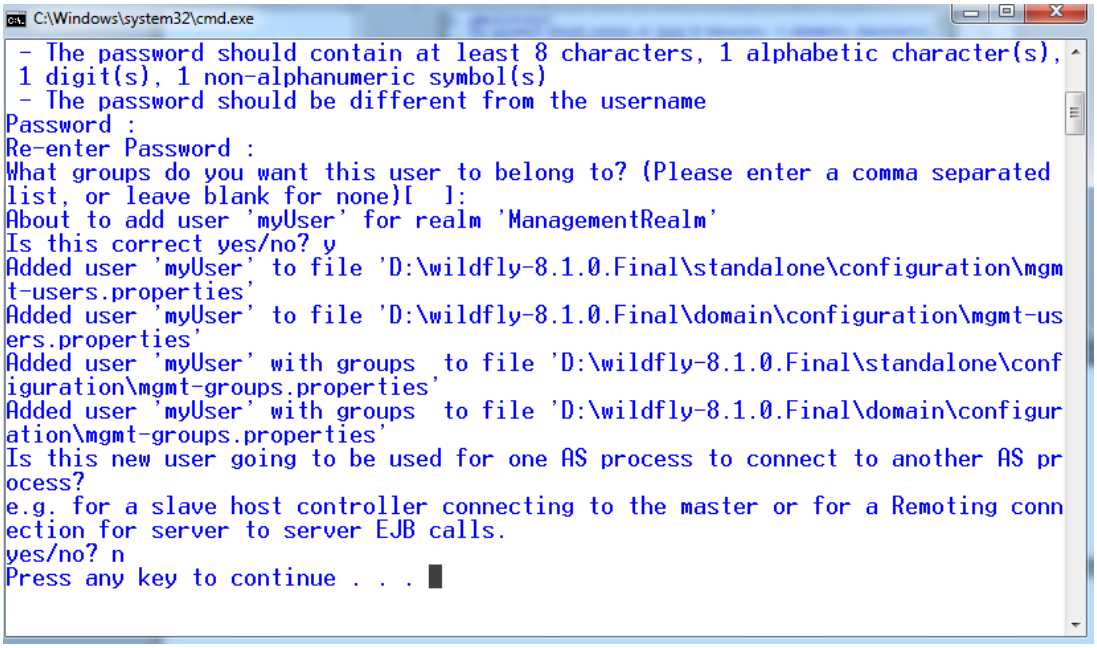

Figure 8

User name: myUser and password: myUser@12.This user will get added under D:\wildfly-8.1.0.Final\standalone\configuration\mgmt-users.properties

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Start the WildFly server.
Open the link: http://localhost:9090/console/.Enter the credentials to log in (Which you have created under Management Realm).

Note: Default port number is 8080. But due to port number conflict, it has been changed to 9090.

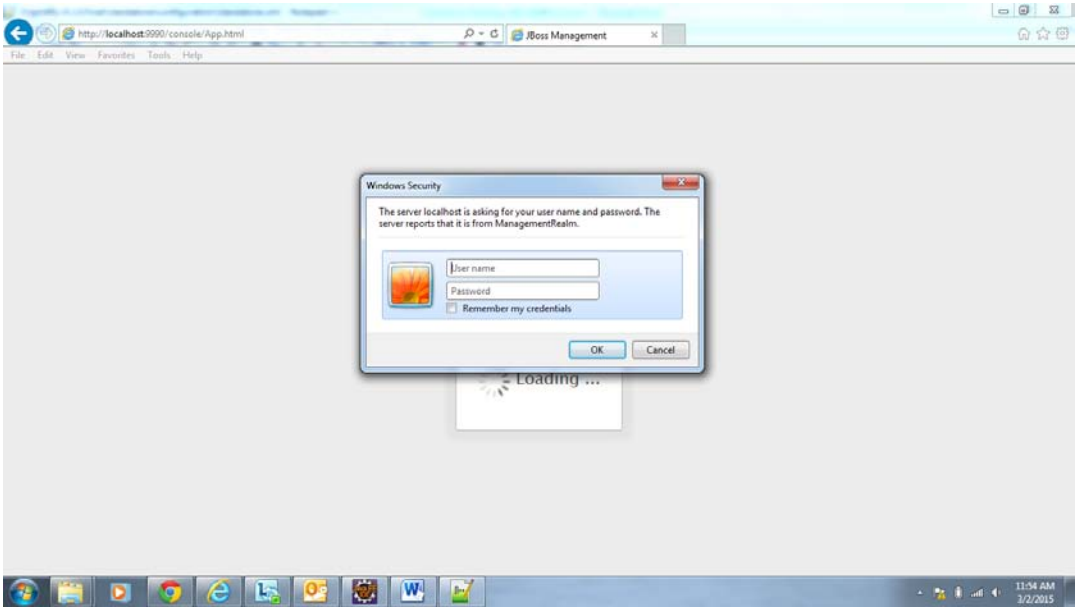

Figure 9

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Following is figure of Home Page

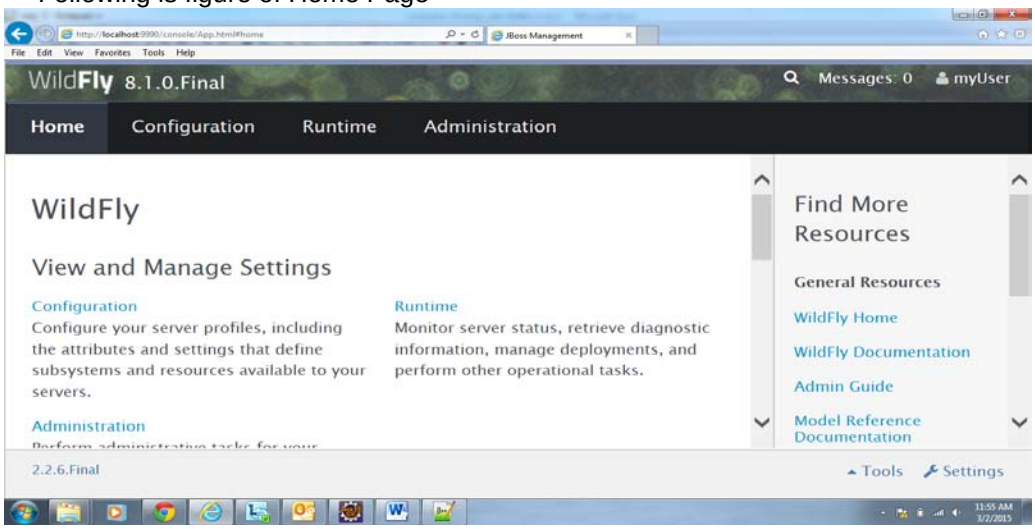

Figure 10

Before creating a datasource we need to deploy the JDBC driver class file to use for connecting to DB.

a. Thus we need to use ojdbc6_g.jar (which is JDBC 4 complaint)
b. Any JDBC4-compliant driver is automatically recognized by WildFly and made available for new datasources.
c. Go to Runtime > Server > Manage Deployments and click on Add to deploy the Oracle complaint Jar file

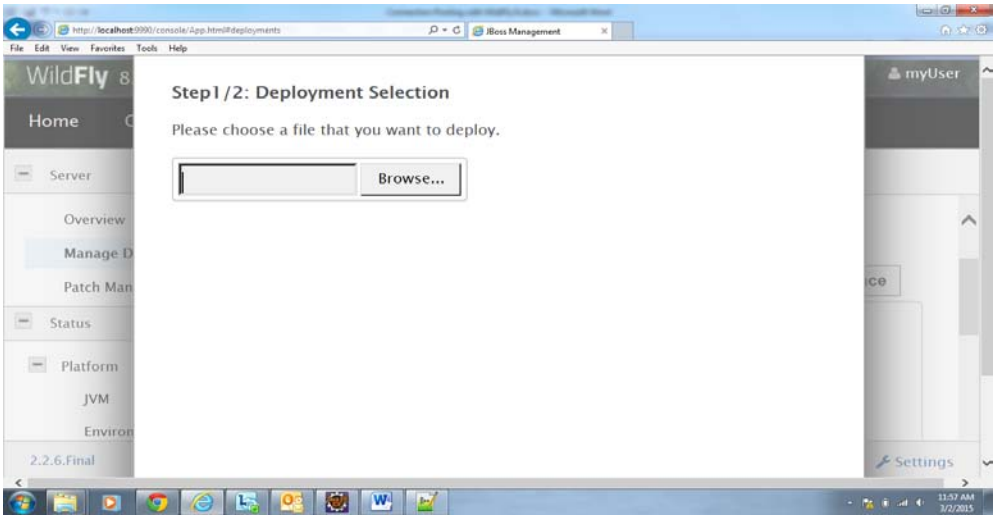

Figure 11

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Give the path from your local machine (where ojdbc6_g.jar) is present

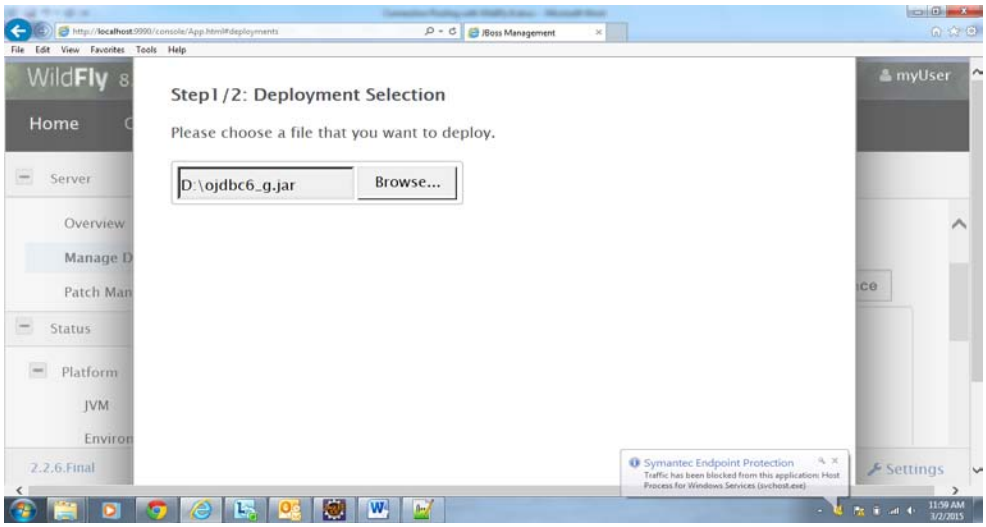

Figure 12

d. Click Next. Supply the Name and Runtime Name with which you want to identify Oracle jar file

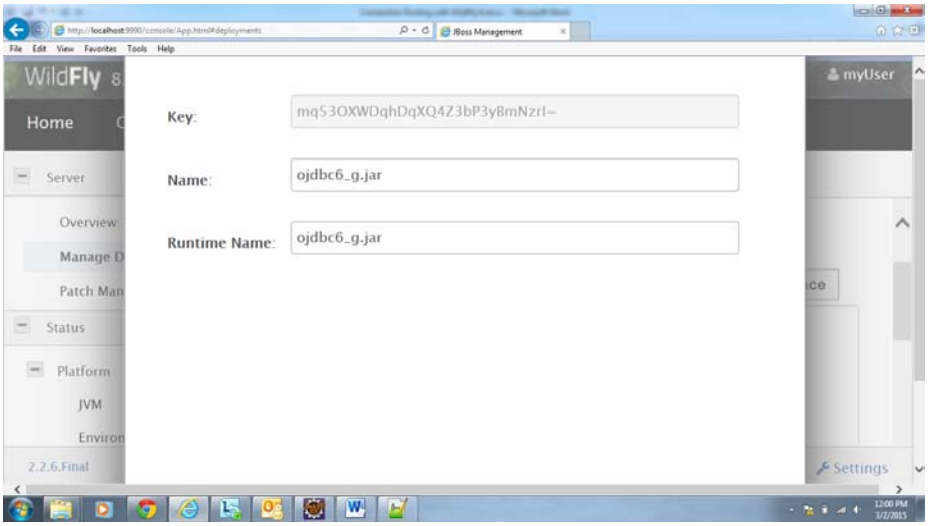

Figure 13

Click on Save

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

e. After Oracle jar file is deployed, click on Enable/Disable to make it available for Datasources

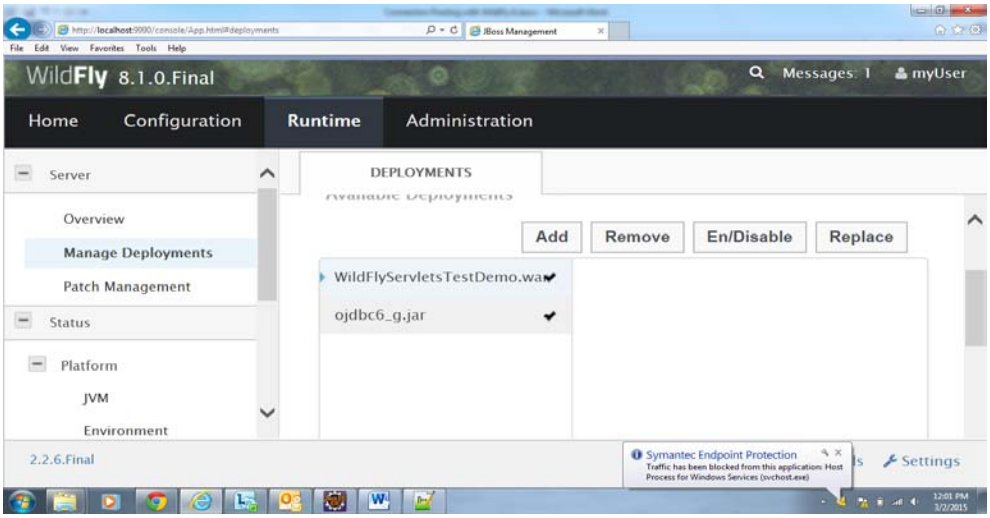

Figure 14

Go to Home > Create a datasource > Datasources. Click on Add to add a Datasource

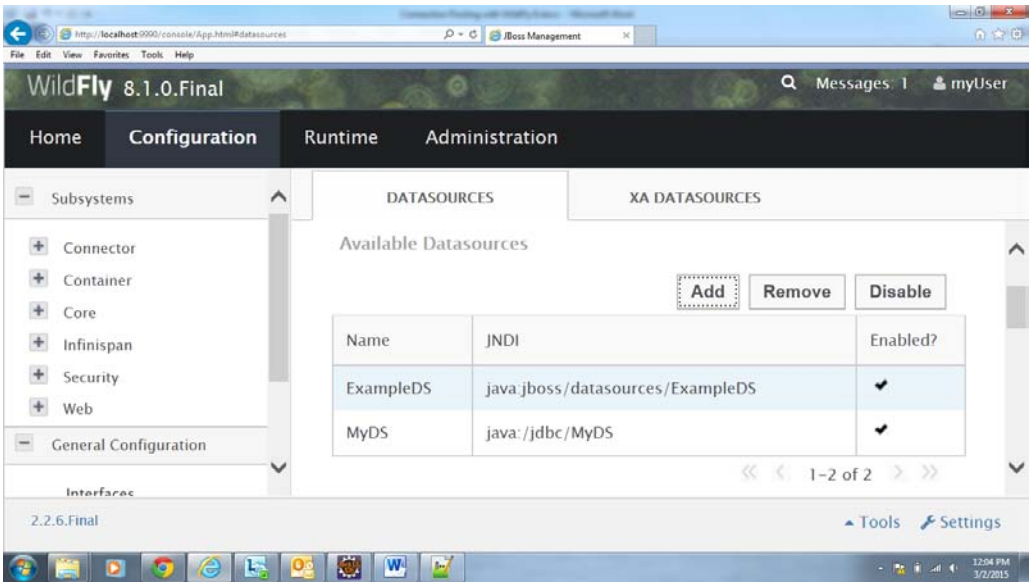

Figure 15

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Supply Connection Pool Name (example: MyDS): as well as JNDI Name: (example: java:/jdbc/MyDS)

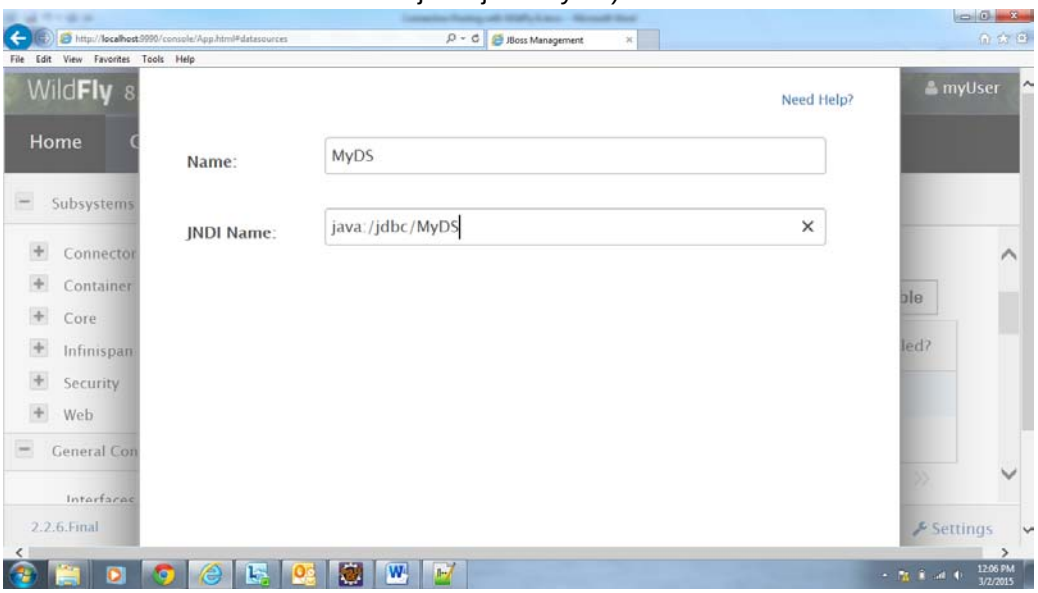

Figure 16

Click Next > Select the Detected Driver > ojdbc6_g.jar

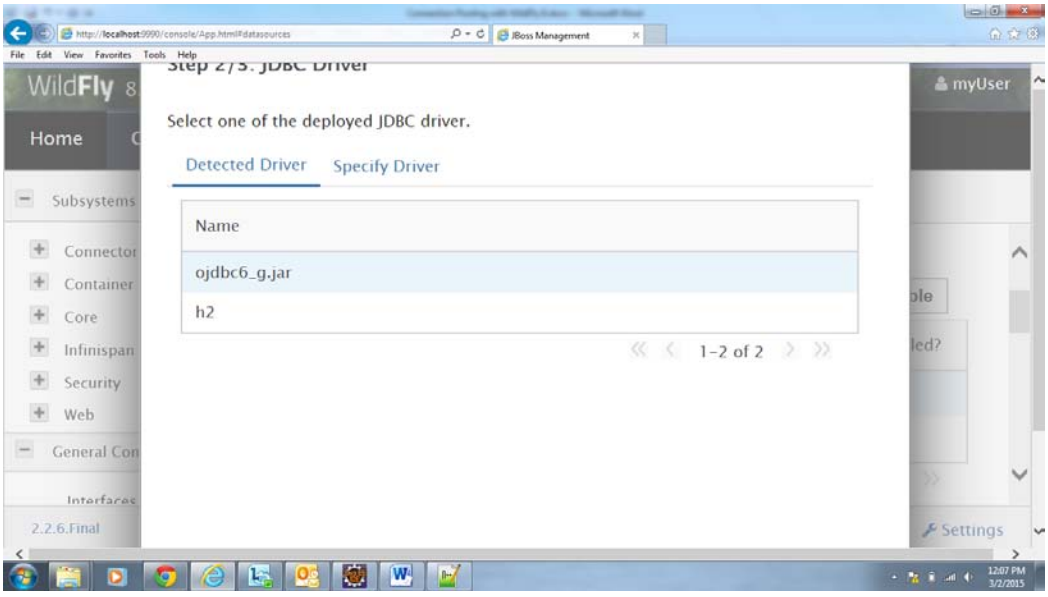

Figure 17

Click Next > Supply Connection URL (jdbc:oracle:thin:@localhost:1522:XE), Username (system), Password (yukti) . These details are with respect to Oracle 11g installed on my machine. These details have to be filled in with respect to the Database you want to connect to.

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

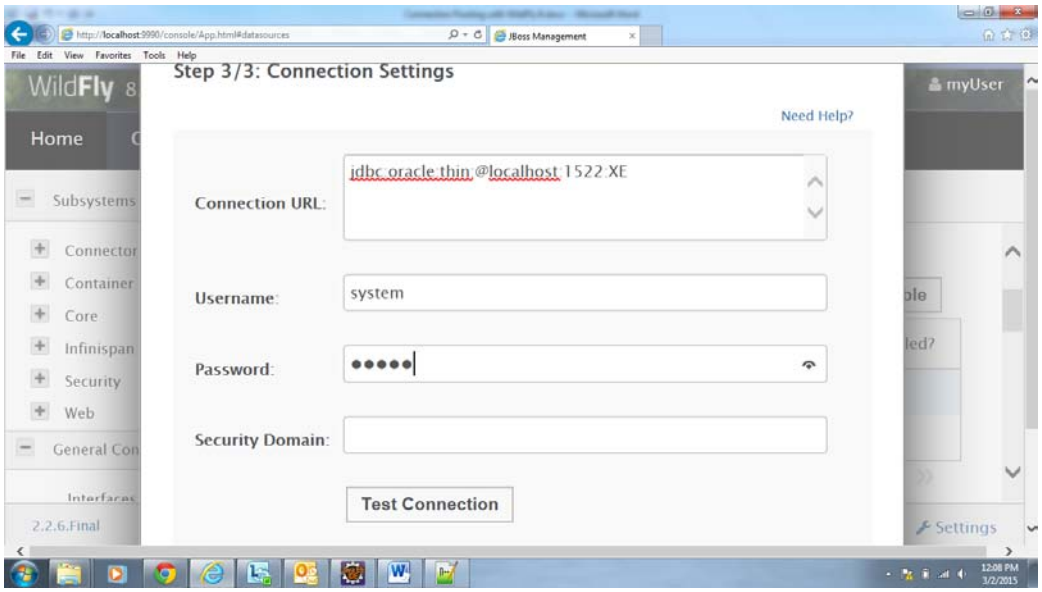

Figure 18

Click on Test Connection. It should show Connection tested successfully

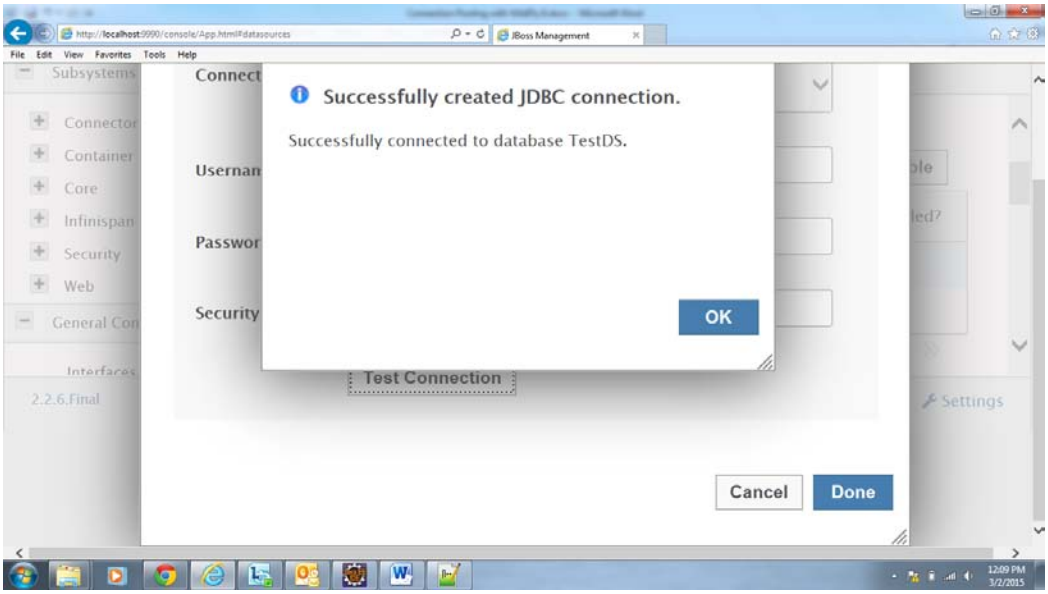

Figure 19

Click on Done to create Datasource.
Select the Datasource and Click on Enable button to make it available for Connection Pooling.

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

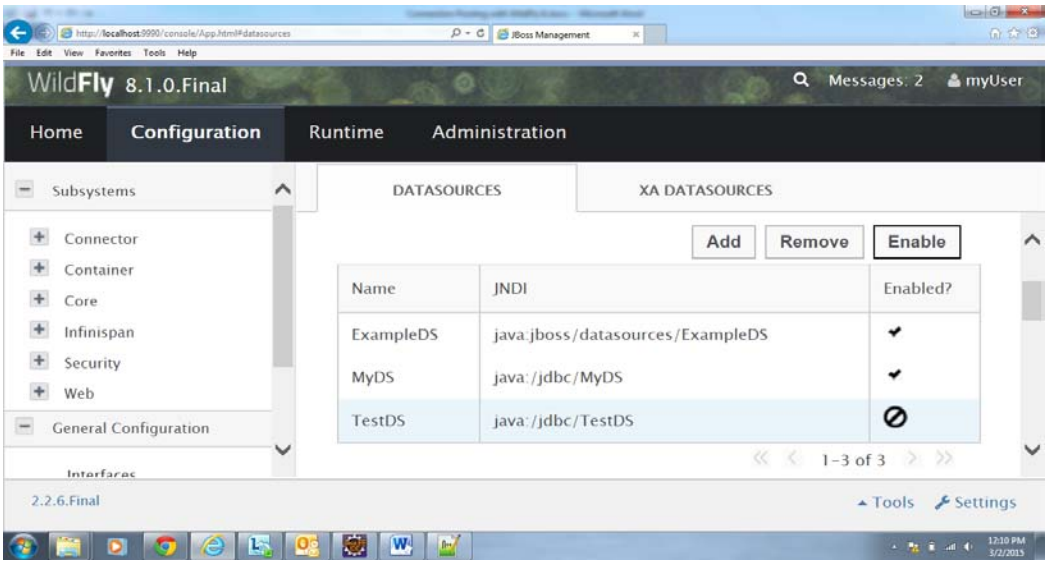

Figure 20

Click on Confirm to Enable Datasource.

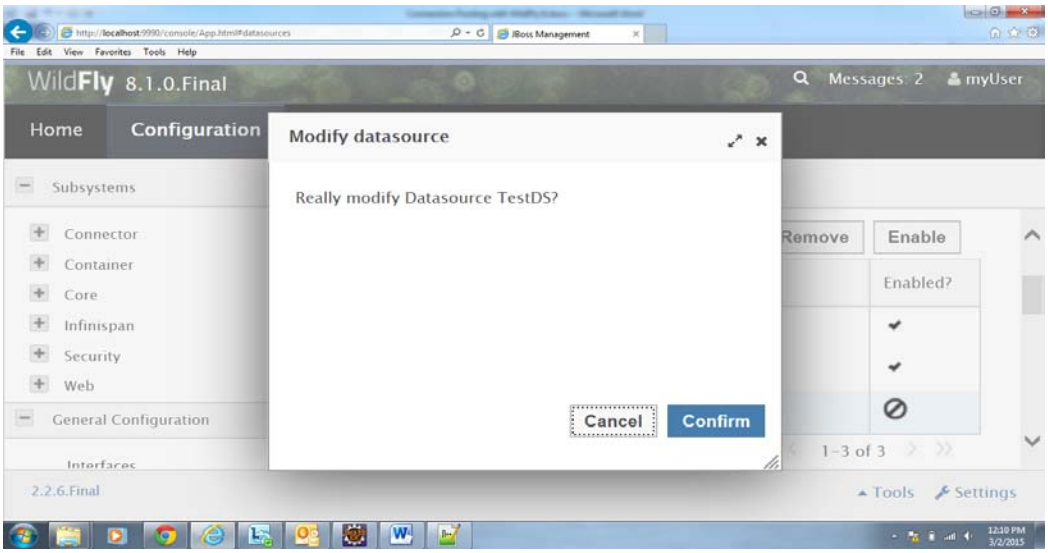

Figure 21

A check mark now appears on Enabled Datasource

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

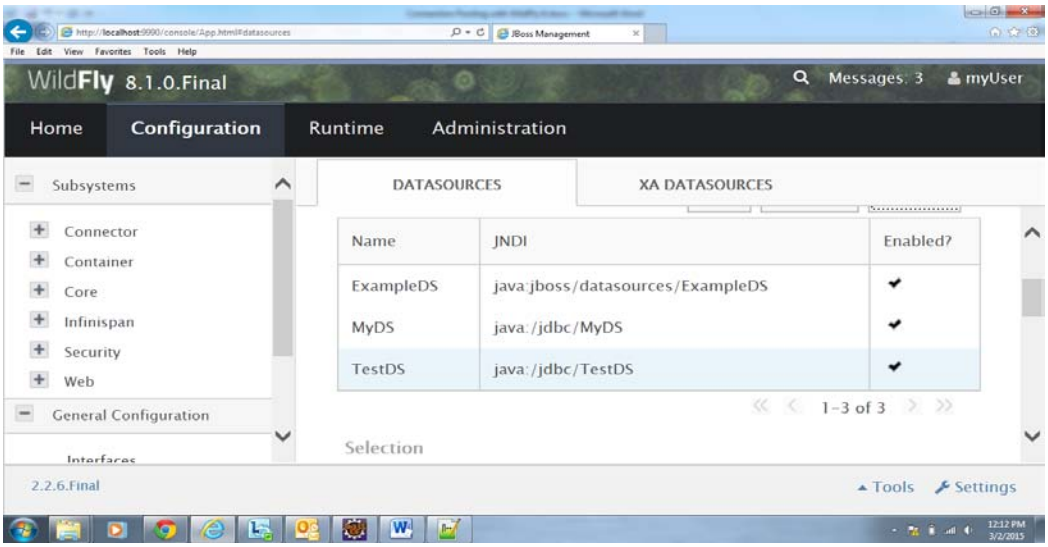

Figure 22

It should now be available on the page of JDBC Datasources
These Datasource mappings would be reflected in standalone.xml file (D:\wildfly-8.1.0.Final\standalone\configuration\standalone.xml). Check Datasource tag in this file.

Write Servlet code to test the connection Pool.

3.2: Design a registration page in HTML. Refer below figure.

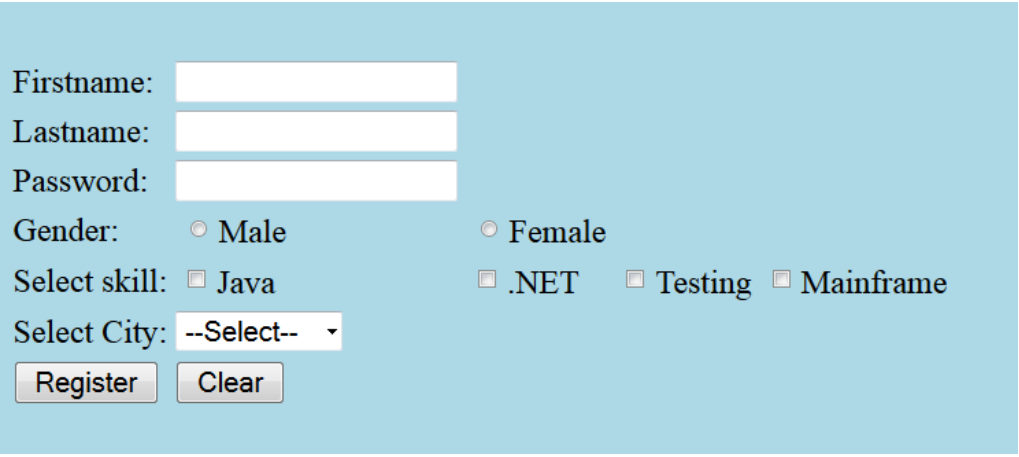

Figure 23

All the details entered in HTML page should be accepted by the Servlet.
The Servlet should connect via Connection Pool to database and store the user registration details in database table.

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

After the user details are inserted successfully in table (RegisteredUsers), display the message "Registration done!!!" in HTML page . Display "Registration failed!!!" in case of any exception .

Note:

- The redirection to HTML page is done using response.sendRedirect (String URLPattern) method
- The application needs to be developed using Layered architecture approach
- User can select more than one skillset, values of which should be concatenated and stored in skillset column

Database Table used: RegisteredUsers

CREATE TABLE RegisteredUsers (firstname VARCHAR (20), lastname VARCHAR (30), password VARCHAR (12) UNIQUE, gender CHAR, skillset VARCHAR (40), city VARCHAR (12));

Refer below figure for layered architecture:

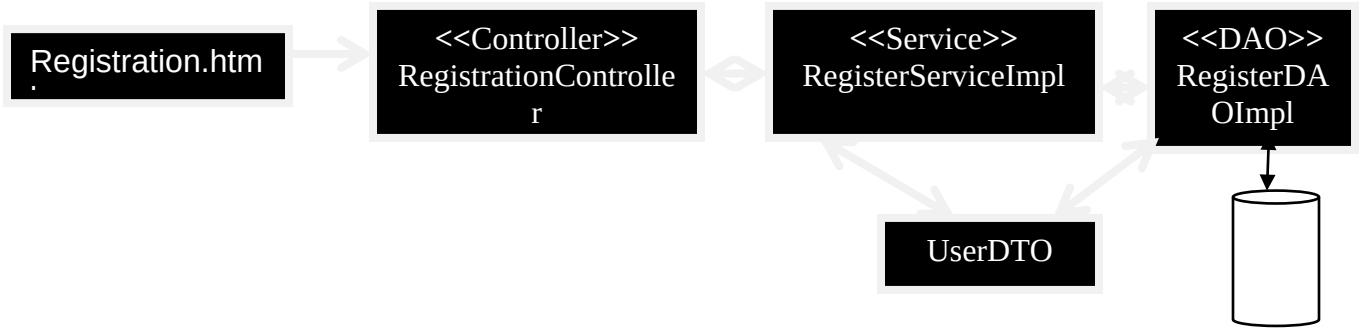

Figure 24

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Lab 4. Inter-Servlet Communication

Goals	• Understanding Inter-Servlet Communication and data transfer between servlets
Time	1 Hour

4.1: Modify the assignment in lab 3.2 and print all the users' registered details on next page.

Note:

- Make use of RequestDispatcher.forward(String URLPattern) method to navigate to next page
- To store all the users' details make use of Request Scope

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Lab 5. Session Management

Goals	• Understanding Session Management
Time	3 Hours

5.1 Rima is working in State electricity board project (eBill application) and is being given the requirement to accept the details online and calculate the net electricity bill amount and persist the details in database.

Refer the below script:

```
CREATE TABLE Consumers(
        consumer_num NUMBER(6) PRIMARY KEY,
        consumer_name VARCHAR2(20) NOT NULL,
        address VARCHAR2(30)
);

INSERT INTO Consumers VALUES(100001,'Sumeet','Shivaji Nagar, Pune');
INSERT INTO Consumers VALUES(100002,'Meenal','M G Colony Panvel,
Mumbai');
INSERT INTO Consumers VALUES(100003,'Neeraj','Whitefield, Bangalore');
INSERT INTO Consumers VALUES(100004,'Arul','Karapakkam, Chennai');

CREATE TABLE BillDetails(
        bill_num NUMBER(6) PRIMARY KEY,
        consumer_num NUMBER(6) REFERENCES Consumers(consumer_num),
        cur_reading NUMBER(5,2),
        unitConsumed NUMBER(5,2),
        netAmount NUMBER(5,2),
        bill_date DATE DEFAULT SYSDATE);

CREATE SEQUENCE seq_bill_num START WITH 100;
```

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

After login the admin user should be able to view the following UI:

[Note : Use login functionality implemented in Lab 2.1]

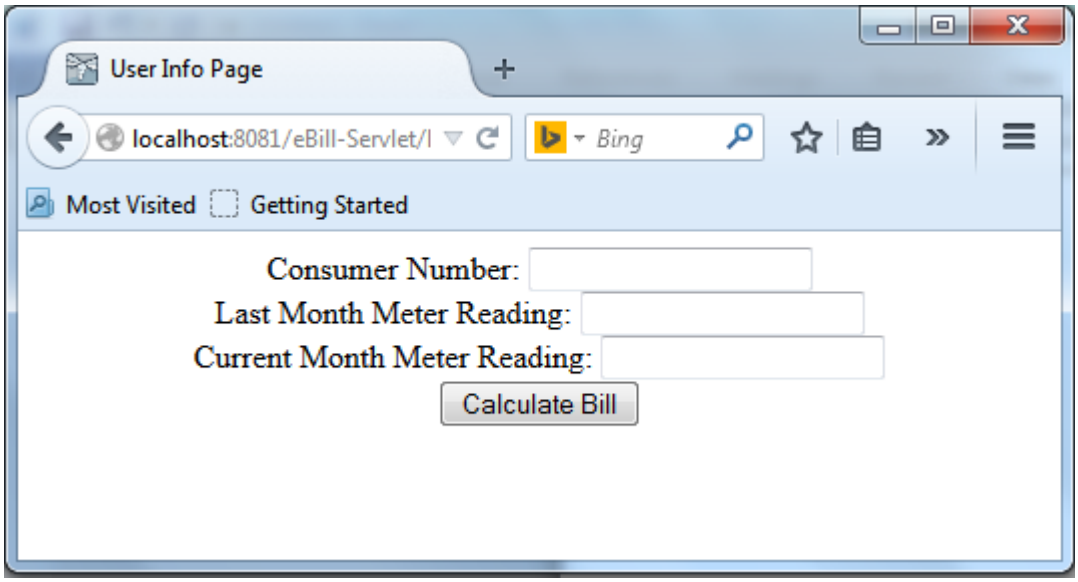

Figure 25

Following Client side validations need to be performed:

- All fields are mandatory
- Consumer number can contain only numbers i.e. 6 digit , Consumer number cannot start with 0
- Last Month and Current Month meter reading can take decimal values, maximum 2 digits after decimal point and should not be negative
- Current Month meter reading cannot be less than Last Month meter reading

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Refer below figure with valid details:

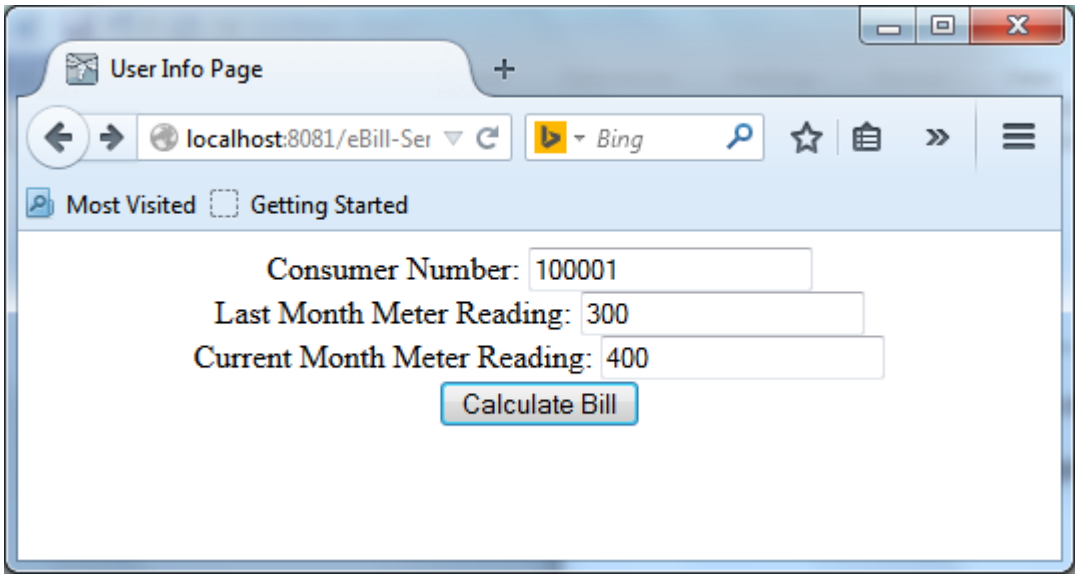

Figure 26

After user enters the valid details Units consumed and Net bill amount need to be calculated in servlet:

Refer below calculations:

Units consumed = (Last month meter reading) – (Current month meter reading)
Net Amount = Unit consumed * 1.15 + Fixed Charge

Assume Fixed Charge is always Rs.100.

After calculating the electricity bill, bill details needs to be inserted into the database table **billdetails.**

Note:

1. Bill_id should be auto generated by sequence.
2. Bill_date should be current date.

Print the following details for the Consumer. Refer below figure.

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

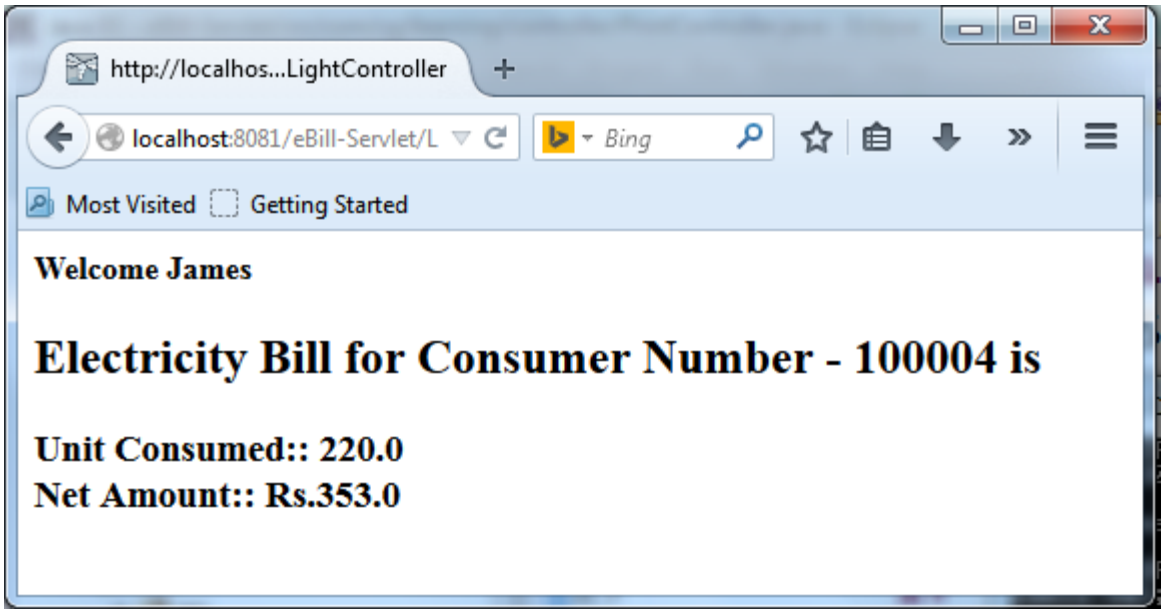

Figure 27

Following figure displayed the error in case of invalid consumer number entered

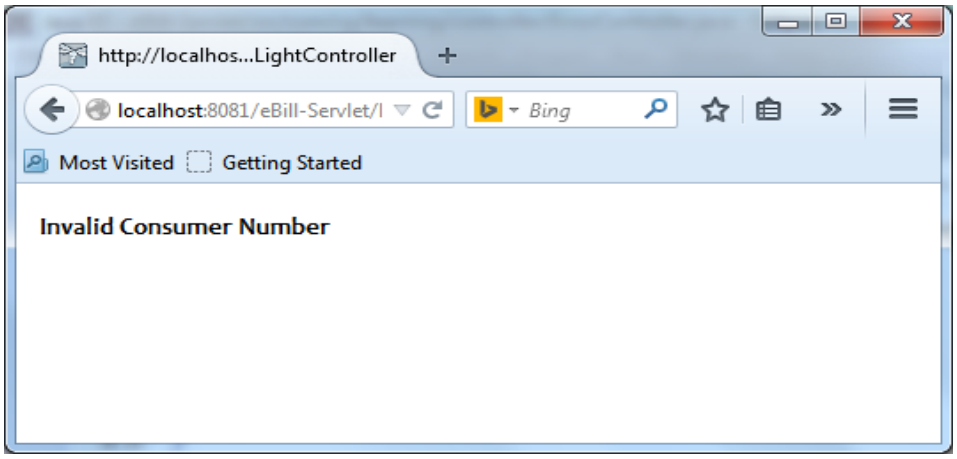

Figure 28

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Layered architecture to be followed:

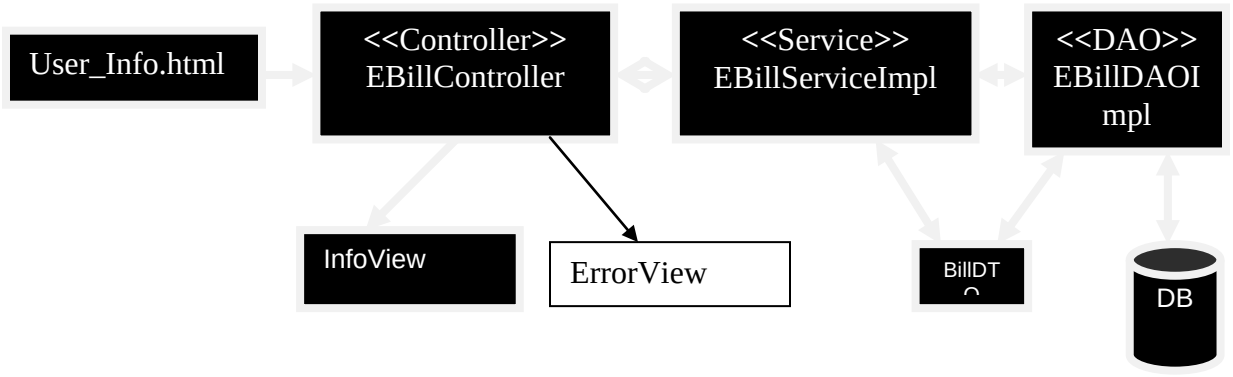

Figure 29

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Lab 6. Filters

Goals	• Understanding the use of filters
Time	0.5 Hour

6.1: Consider below filter code snippet, many servlet requests are passing through this Filter. If servlets are invoked then predict the output.

```
public void doFilter(ServletRequest request, ServletResponse response,
      FilterChain chain) throws IOException, ServletException {
   |
   long before = System.currentTimeMillis();
   chain.doFilter(request, response);
   long after = System.currentTimeMillis();
   String name = "";
   if (request instanceof HttpServletRequest) {
     name = ((HttpServletRequest)request).getRequestURI();
   }
   config.getServletContext().log(name + ": " + (after - before) + "ms");
}
```

Figure 30

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Additional Exercise: <<To Do>>

Understanding Session Management with Cookies:

Implementing a **Remember Me** assignment with **cookies**. A user wants to login to an application, but first is being presented with a welcome page. The welcome page has a “remember me” check box. When user clicks on checkbox and says “remember me” the next time when he access the application he / she should be shown the Login Page directly and not Welcome Page. If the user does not click on “remember me” checkbox then next time when he access the application he / she should be shown the Welcome Page and on click of contine button display the Login Page.

See below code snippets and fill in the blank spaces left with cookie / servlet related code.

Welcome.html looks as below

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="ISO-8859-1">
5  <title>Welcome Page</title>
6  </head>
7  <body bgcolor="gold" style="text-align: center">
8  <form action="NextPage" method="get">
9     <input type="submit" value="Continue"/><BR>
10    <input type="checkbox" name="remember"/>Don't show this page again
11 </form>
12 </body>
13 </html>
```

Figure 31

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Note: This servlet has to be the first page to execute to check for the availability of cookies. CheckServlet.java code snippet as below: Here from doGet (request, response) method doPost (request, response) is invoked

```
protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
    // code to obtain the existing cookies

    //checking if cookies are present)
    {
        // if cookies - not present then redirecting to welcome page
        response.sendRedirect("welcome.html");
    }
    // if cookies present, then
        // code to iterate over cookies

        // code to obtain the cookie name and value pair

        // code to check if cookie name = remember and cookie value is yes i.e check box checked ir

        // if cookie value is remember me then code to redirect to login page
```

Figure 32

Refer NextPage Servlet below to create a cookie.
Here from doPost (request, response) method doGet (request, response) is invoked.

```
protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
    // checking if check box is selected
    String status = request.getParameter("remember");
    if (status != null) {
        // code to create a cookie, and setting age of cookie to 160 seconds
        // and adding cookie to response



    }
    // redirecting to login page
    response.sendRedirect("login.html");
}
```

Figure 33

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Below is login.html page.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body bgcolor="gold" style="text-align: center">
<form>
User Name: <input type="text" name="name"/><BR>
Password: <input type="password" name="pwd" /><BR>
<input type="submit" value="Login"/>
</form>
</body>
</html>
```

Figure 34

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.

Appendix A: Table of Figures

©2016 Capgemini. All rights reserved.
The information contained in this document is proprietary and confidential. For Capgemini only.