



­

Assignment Document:

Core Spring

Version: Core Spring Practice CaseStudy 2/ASSIGNMENT/1.0

Date: 11-05-2015

Cognizant

500 Glen Pointe Center West

Teaneck, NJ 07666

Ph: 201-801-0233  
[www.cognizant.com](http://www.cognizant.com)

Table of Contents

[Cognizant Higher Education Support 3](#_Toc475627374)

[1. Infrastructure Section 3](#_Toc475627375)

[i. Hardware, Software Specification 3](#_Toc475627376)

[2. Case Study Assignments 5](#_Toc475627377)

[3. Pre- requisite steps 5](#_Toc475627378)

[4. Database Setup 5](#_Toc475627383)

[5. Register Higher Education Course 6](#_Toc475627386)

[6. Technical Specifications 6](#_Toc475627387)

[7. Business Rules& Validations 7](#_Toc475627388)

[8. Sequence Diagram 10](#_Toc475627389)

[9. View Registered Course Details 10](#_Toc475627390)

[10. Technical Specifications 11](#_Toc475627391)

[11. Business Rules & Validations 12](#_Toc475627392)

[12. Sequence Diagram 12](#_Toc475627393)

[13. Solution 12](#_Toc475627394)

[14. Evaluation Rubrics 13](#_Toc475627395)

[15. Summary of this Case Study 13](#_Toc475627396)

# Cognizant Higher Education Support

1. Infrastructure Section
2. Hardware, Software Specification

This section captures the hardware and software specifications for the effective delivery of the course.

1. **Hardware Specification**

|  |  |
| --- | --- |
| **Server Specification** | Windows 7 - 32bit  4GB RAM |
| **Desktop/Client Specification** | Admin Rights |

1. **Software** **Specification**

| **#** | **Name of the software(s) to be installed** | **Version** | **License available in RAMS?**  **(Yes/No)** | **License available in RAMS for onsite?**  **(Yes/No)** | **Description of any patch to be installed** | **Can be used through Tool Wire / SoftGrid?** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | JDK | 7.0 | Yes | Yes |  | Yes |
| 2 | SDE 7.0 | 7.0 | Yes | Yes |  | Yes |
| 3 | Tomcat 6.0/7.0 | 6.0/7.0 | Yes | Yes |  | Yes |
| 4 | MySQL Workbench | 6.0.8 | Yes | Yes |  | Yes |
| 5 | Netbeans 6.9.1- Complete Installation (Including Derby database and Tomcat Server) | 6.9.1 | Yes | Yes |  | Yes |
| 6 | Eclipse | 3.6 | Yes | Yes |  | Yes |

**Instruction for installing the software in the “Tool Wire/SoftGrid” environment (if the response is ‘Yes’ in the last column in the above table):**

| **#** | **Name of the software** | **Instruction** |
| --- | --- | --- |
| 1 | Spring Jars 3.0 | <http://www.java2s.com/Code/Jar/s/Downloadspringweb310RELEASEjar.htm> |
| 2 | Mysql-connector-java-5.1.12-bin | <http://www.java2s.com/Code/Jar/m/Downloadmysqlconnectorjava5123binjar.htm> |
| 3 | Spring-hibernate3-2.0.8 | <http://www.java2s.com/Code/Jar/s/Downloadspringhibernate3208jar.htm> |
| 4 | Hibernate 3.5 jars | <http://www.java2s.com/Code/Jar/h/Downloadhibernate353jar.htm> |

**Installation Details:**

Refer the below link for JDK and JRE installation:

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html>

Refer the below link for SE download and installation:

<https://gto.cognizant.com/javacoe/SitePages/SDEDownload.aspx>

**Test Cases to check the installed software:**



1. Case Study Assignments

Estimated Completion Time: xx Minutes

Objective:

Cognizant Higher Education Support (HES) encourages employees for continuing education through distance learning/part time programs in India with reputed academic institutions, or certification and Ph.D. programs from organizations of international repute. They want to automate the process of registering with the HES council.

They would like to develop the below components:

* + **Service1: Registration for Higher Education Course**
  + **Service2: View Registered Course details**

**Control flow is Tester --> Manager --> Façade --> BO --> Dao**

1. Pre- requisite steps

* Please make sure that JAVA\_HOME is set your JDK installation folder, otherwise please set as following: *C:\Program Files\Java\jdk1.7.0\_51*
* Unzip the below content and place it in your hard disk



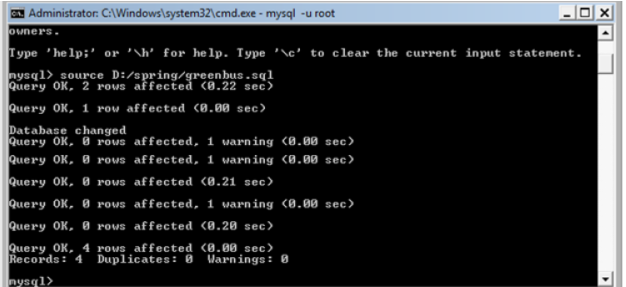
* Open Eclipse IDE or SDE and import the below project using “Import -> Existing Projects into workspace” option
* This acts as your code base to begin the case study development

1. Database Setup

* Unzip the below content and place it in your hard disk



* Open MySQL Command Line and run the script using the command:
  + Example: source D:/Javaca/library.sql (after unzipping the sql file please run this command by giving the location where the .sql file is placed)
  + Queries should execute without any errors



Note: if you want to drop and re-create any tables or re-insert data for testing purposes, please run the same .sql script again as given in step 2 to reset your database setup.

1. Register Higher Education Course

The service is used by the system to register the employee for Higher Education Course and compute the salary for each associate. This service is going to become a part of different other modules in the system and will be developed as an independent module, so that it can be plugged into other modules easily.

This service accepts the employee details and registers them for Higher Education program. If the above business rules are satisfied, then persist the data and will return an Integer value (registrationId). If any of the business rules is not satisfied, throws an appropriate user defined exception as given in the below ‘Technical Specifications’ table.

1. Technical Specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Method Name** | **Input** | **Output** | **Exception** |
| HESManager | **registerCourse()** | **RegistrationVO registrationVO** | Integer registrationId | InvalidJobLevelException InvalidCollegeIdException  InvalidModeOfStudyException  These exceptions have to be caught and thrown back to Tester class |
| HESFacade | **registerCourse()** | **RegistrationVO registrationVO** | Integer registrationId | InvalidJobLevelException InvalidCollegeIdException  InvalidModeOfStudyException  These exceptions have to be caught and thrown back to HESManager class |
| HESBO | **registerCourse()**  \*\* check for the following business rules given below | **RegistrationVO registrationVO** | Integer registrationId | InvalidJobLevelException InvalidCollegeIdException  InvalidModeOfStudyException  These exceptions have to be caught and thrown back to HESFacade class |
| HESDao interface | **(Integer collegeId)**: | Integer CollegeId |  |  |
| **registerCourse()** | **RegistrationVO registrationVO** |  |  |
| HESDaoImpl  implements HESDao | **getCollegeDetails (Integer collegeId)**: | Integer CollegeId | return the corresponding College object for the collegeId if valid, null if invalid |  |
| **registerCourse()** | **RegistrationVO registrationVO** | Integer **registrationId** in case of successful execution |  |
| hes\_config.xml | Contains all the xml configurations related to Service 1 |  |  |  |

1. Business Rules& Validations

In the **HESBO class**,check for the following business rules:

|  |  |  |
| --- | --- | --- |
| **Rule**  **No.** | **Rule Description** | **User Define Exception to be thrown** |
| 1. | **jobLevel** entered should not be negative. It can be a minimum value of 3 and maximum value of 7. | If jobLevel is invalid, throws  **InvalidJobLevelException** |
| 2. | **collegeId** entered should be valid [getCollegeDetails(Integer collegeId) method of HESDao. Method returns College object if collegeId is valid and null if not valid]. | If collegeId is invalid, throws  **InvalidColegeIdException** |
| 3. | **modeOfStudy** entered should be ‘P’ or ‘D’. ‘P’ denotes PartTime and ‘D’ denotes DistanceEducation. | If modeOfStudy is invalid, throws  **InvalidModeOfStudyException** |

If any of the business rules is not satisfied, throw an appropriate user defined exception as given in the above table.

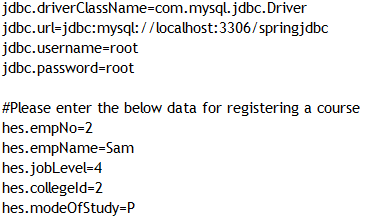
**Note:** The RegistrationVO, College, Registration class will be provided with the required getter and setter methods which should not be modified. No changes should be done to the skeletons provided (Especially the name and method signature).

**Limitations and Constraints**

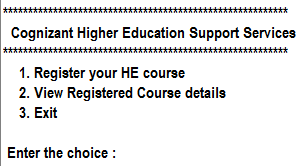
1. All the java classes should be declared as beans in spring configuration file hes\_config.xml.
2. HESFacade should be integrated in HESManager using setter-based dependency injection.
3. HESBO should be integrated with HESFacade using constructor-based dependency injection.
4. HESDao should be integrated with HESDao using setter-based dependency injection.
5. Use ONLY **hes\_config.xml** for all configurations related to Service1(**Register Higher Education Course).**
6. Read the **input data** and **Data Source** details from the **constants.properties.**
7. Enable transactions by configuring transaction manager.

**<SQL Connection details will be provided to the associate as below>**

Create a **constant.property** file with the following details. Please set the value for DBName and password, and also set the values for registering a course.

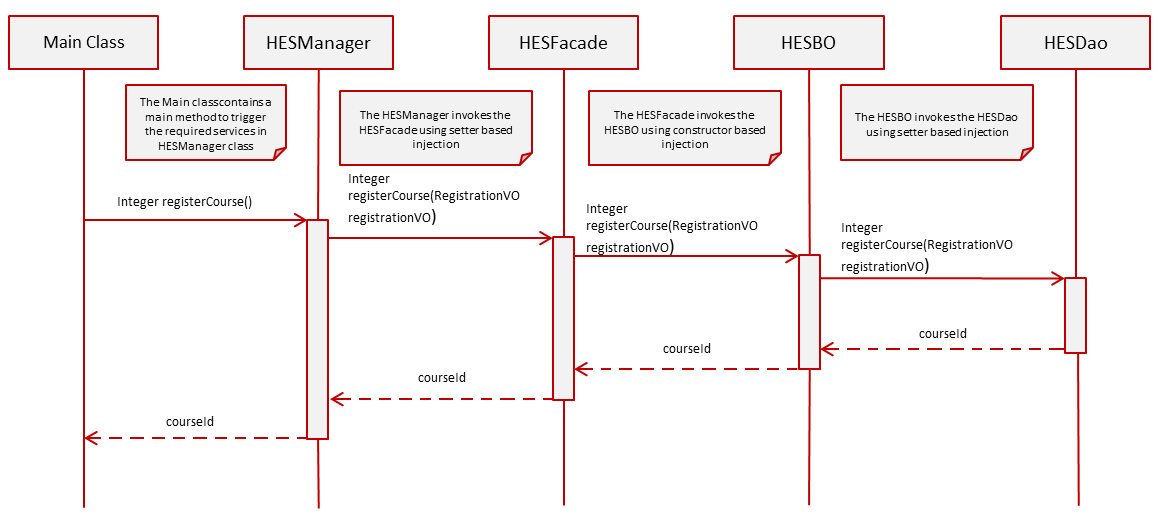


**Output:**



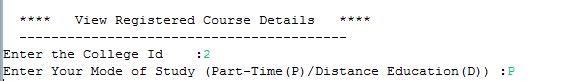


1. Sequence Diagram

****

1. View Registered Course Details

* When the user selects Option 2**, viewRegisteredDetails() method of the Tester** is invoked.
* The HES admin will input the **collegeId and modeOfStudy.**
* The **ViewRegisteredCourseDetails** menu in the **viewRegisteredDetails()** method of the Tester class looks like below:

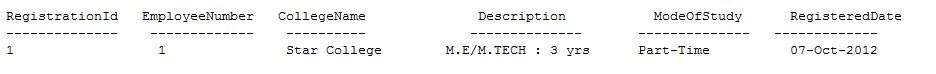


The control flow is **Tester --> Manager --> Façade --> BO --> Dao**

In the **Tester** Class,

* Invoke **viewRegisteredDetails (Integer collegeId, String modeOfStudy)** from Tester to **HES**Manager following the control flow is mentioned above. This method returns a list of type **DetailsVO** objects.
* Write code to display the output in the below format by using the returned values, if the business rules are satisfied, otherwise display the appropriate exception message
* **Hint:** To display registrationDate in proper Date Format (dd-MMM-yyyy), use **DateUtility.getStringFromDate (Date registrationDate)** to convert registrationDate from Date to String.
* To display ModeOfStudy in proper Format (Part Time (P) or Distance Education (D)), use **ModeConverter.getMode (String modeOfStudy)** to convert modeOfStudy denotes Part Time (P) or Distance Education (D).

**OUTPUT**



1. Technical Specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Method Name** | **Input** | **Output** | **Exception** |
| HESManager | **viewRegisteredDetails()** | Integer collegeId  String modeOfStudy | List<DetailsVO> detailVoList | NoRecordException  This exception has to be caught and thrown back to Tester class |
| HESFacade | **viewRegisteredDetails()** | Integer collegeId  String modeOfStudy | List<DetailsVO> detailVoList | NoRecordException  This exception has to be caught and thrown back to HESManager class |
| HESBO | **viewRegisteredDetails()** | Integer collegeId  String modeOfStudy | List<DetailsVO> detailVoList | NoRecordException  This exception has to be caught and thrown back to HESFacade class |
| HESDAO Interface | **viewRegisteredDetails()** | Integer collegeId  String modeOfStudy | List<DetailsVO> detailVoList |  |
| HESDAOImpl | **viewRegisteredDetails()** | Integer collegeId  String modeOfStudy | List<DetailsVO> detailVoList |  |
| hes\_config.xml | Contains all the xml configurations related to Service 1 |  |  |  |

1. Business Rules & Validations

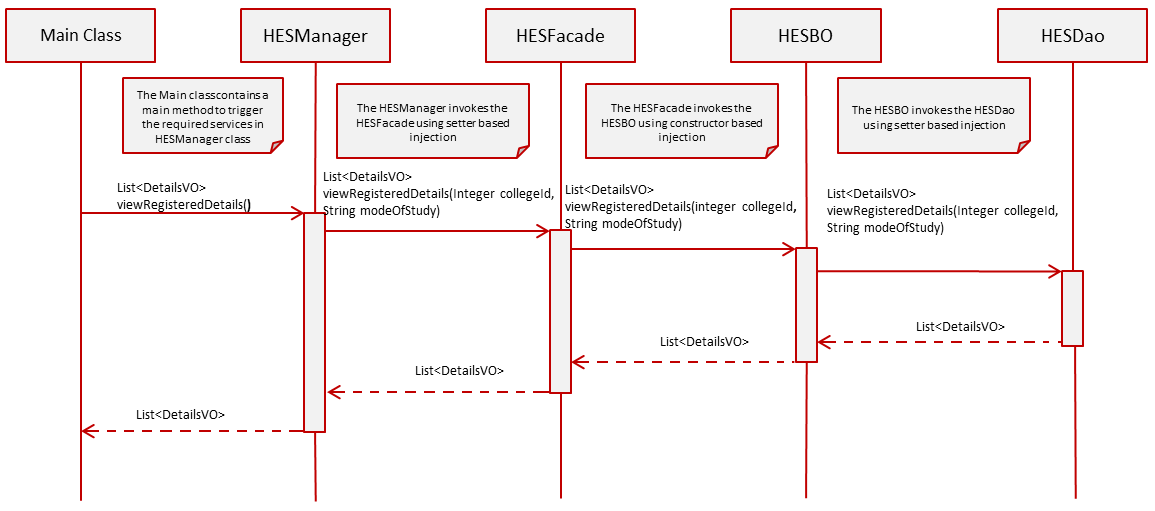
In the **HESBO class** check for the following business rules:

|  |  |  |
| --- | --- | --- |
| **Rule**  **No.** | **Rule Description** | **User Define Exception to be thrown** |
| 1. | List<DetailsVO> returned by the method viewRegistrationDetails() of HESDao class should not be empty. | If list is empty  **NoRecordException** |

If any of the business rules is not satisfied, throw an appropriate user defined exception as given in the below table.

**Note:** The DetailsVO class will be provided with the required getter and setter methods which should not be modified. No changes should be done to the skeletons provided (especially to the name and method signature.)

1. Sequence Diagram

****

1. Solution



1. Evaluation Rubrics

|  |  |
| --- | --- |
| Parameters | Weightage |
| 1. Completeness |  |
| 1. Accuracy |  |
| 1. Clarity of understanding |  |
| 1. Presentation |  |
| Total |  |

1. Summary of this Case Study

You have just learnt:

* Spring framework and IOC container
* Spring dependency injection
* Spring JDBC framework
* Spring transaction management
* ResourceBundleMessageSource