



­

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

[Marriage Hall Booking 2](#_Toc452460122)

[1. Infrastructure Section 2](#_Toc452460123)

[i. Hardware, Software Specification 2](#_Toc452460124)

[2. Case Study Assignments 3](#_Toc452460125)

[3. Pre – requisite steps………………………………………………………………………......4](#_Toc452460127)

[4. Database setup………. 4](#_Toc452460127)

[5. Book Wedding Hall 5](#_Toc452460128)

[6. Technical Specifications 5](#_Toc452460129)

[7. Business Rules& Validations 6](#_Toc452460130)

[8. Sequence Diagram 9](#_Toc452460131)

[9. View Hall Booking Details 9](#_Toc452460132)

[10. Technical Specifications 10](#_Toc452460133)

[11. Business Rules & Validations 10](#_Toc452460134)

[12. Sequence Diagram 11](#_Toc452460135)

[13. Solution 11](#_Toc452460136)

[14. Evaluation Rubrics 12](#_Toc452460137)

[15. Summary of this Case Study 12](#_Toc452460138)

**Marriage Hall Booking**

**Infrastructure Section**

1. **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:**



**Case Study Assignments**

Estimated Completion Time: xx Minutes

Objective:

ABC Wedding Halls has several hotels with a luxurious wedding banquet hall. Chic, contemporary, and vibrant, it caters to the wedding needs. They are manually managing the booking of the marriage hall, whenever customers book. They may also need to view the booking details of a particular hall. They want to automate the process of booking the marriage halls and viewing the booking details.

They would like to develop the below components:

**Service1:** Book wedding halls

**Service2:** View booking details of the halls

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

**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



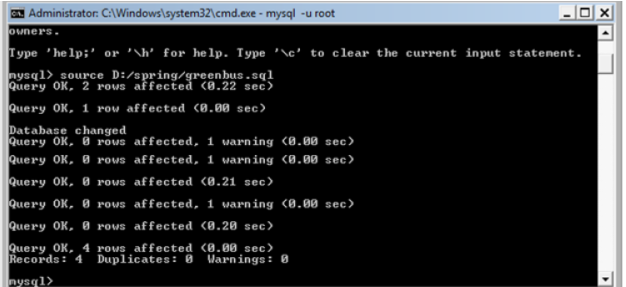
* Open Eclipse IDE or SDE and import the below project using “Import -> Existing Projects into workspace” option
* This acts as your code base to begin the case study development

**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.

**Book Wedding Hall:**

The service allows the Customers to book the wedding hall. This service is going to become a part of different other modules in the system and hence will be developed as an independent module, so that it can be plugged into other modules easily.

This service accepts the customer details and book the Marriage Hall. If the business rules are satisfied, then persists the data and returns an Integer value (registrationId). If any of the business rules is not satisfied, throws an appropriate user defined exception as given in the below table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Method Name** | **Input** | **Output** | **Exception** |
| MarriageHallManager | bookHall() | BookingVO bookingVO | Integer bookingId | InvalidHallNoException InvalidCustomerNameException  InvalidBookingDateException  InvalidFromToDateException  InvalidPhoneNoException  InvalidAvailabilityException    These exceptions have to be caught and thrown back to Tester class. |
| MarriageHallFacade | bookHall() | BookingVO bookingVO | Integer bookingId | InvalidHallNoException InvalidCustomerNameException  InvalidBookingDateException  InvalidFromToDateException  InvalidPhoneNoException  InvalidAvailabilityException  These exceptions have to be caught and thrown back to Tester class. |
| MarriageHallBO | bookHall()  \*\* check for the following business rules given below | BookingVO bookingVO | Integer bookingId | InvalidHallNoException InvalidCustomerNameException  InvalidBookingDateException  InvalidFromToDateException  InvalidPhoneNoException  InvalidAvailabilityException  These exceptions have to be caught and thrown back to Tester class. |
| MarriageHallDAO interface | getHallDetails () | Integer hallNo |  |  |
| bookHall() | BookingVO bookingVO |  |  |
| MarriageHallDAOImpl | getHallDetails () | Integer hallNo | return the corresponding HallDetails object for the hallNo if valid, null if invalid |  |
| bookHall() | BookingVO bookingVO | Integer bookingId in case of successful execution |  |
| hall\_config.xml | Contains all the xml configurations related to Service1(Book Wedding Hall) |  |  |  |

**Business Rules& Validations:**

In the MarriageHallBO class check for the following business rules:

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule Description** | **User Define Exception to be thrown** |
| 1. | HallNo entered should be valid [use getHallDetails (Integer hallNo) method of MarriageHallDAO. Methods return HallDetails object, if hallNo is valid and null if invalid]. | If empNo is invalid, throws  InvalidHallNoException. |
| 2. | customerName entered should contain only alphabets and spaces and should not contain any digits and special characters. | If empName is invalid, throws InvalidCustomerNameException. |
| 3. | fromDate entered should be valid [use checkFromToDate (Calendar fromDate) method of CalendarUtility. Methods return boolean value as true, if fromDate is valid and false if invalid]. | If fromDate is invalid, throws InvalidBookingDateException. |
| 4 | fromDate and toDate entered should be valid [use checkValidDate (Calendar fromDate, Calendar toDate) method of CalendarUtility. Methods return boolean value as true if fromDate and toDate is valid and false if invalid] | If fromDate and toDate is invalid, throws InvalidFromToDateException. |
| 5 | phoneNo entered should contain only 10-digit numbers | If phoneNo is invalid, throws InvalidPhoneNoException. |
| 6 | Check the availability of the hall [use checkAvalibility (BookingVO bookingVO) method of MarrigeHallDao. Methods return list of BookingDetails, if hall is already booked within the time period fromDate to toDate, null if booking not available] | If list of Booklist is greater than zero, throws InvalidAvailabilityException. |

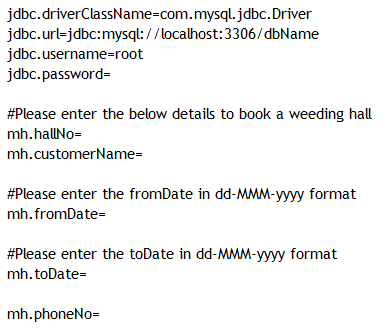
**Note:** The BookingVO, BookingDetails, HallDetails class will be provided with the required getter and setter methods which should not be modified. No changes should be done to the skeletons provided (Especially the name and method signature).

**Limitations and Constraints**

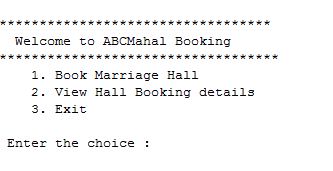
* All the above mentioned java classes are to be declared as beans in spring configuration file.
* MarriageHallFacade should be integrated MarriageHallManager using Setter Based Dependency Injection.
* MarriageHallPortBO should be integrated with MarriageHallFacade using Constructor based dependency injection.
* MarriageHallDAO should be integrated with MarriageHallBO using Setter Based Dependency.
* Use ONLY hall\_config.xml for all configurations related to Service1(Book Wedding Hall).
* Read the input data and Data Source details from the constants.properties.
* 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, password and also set the values to book a wedding hall.

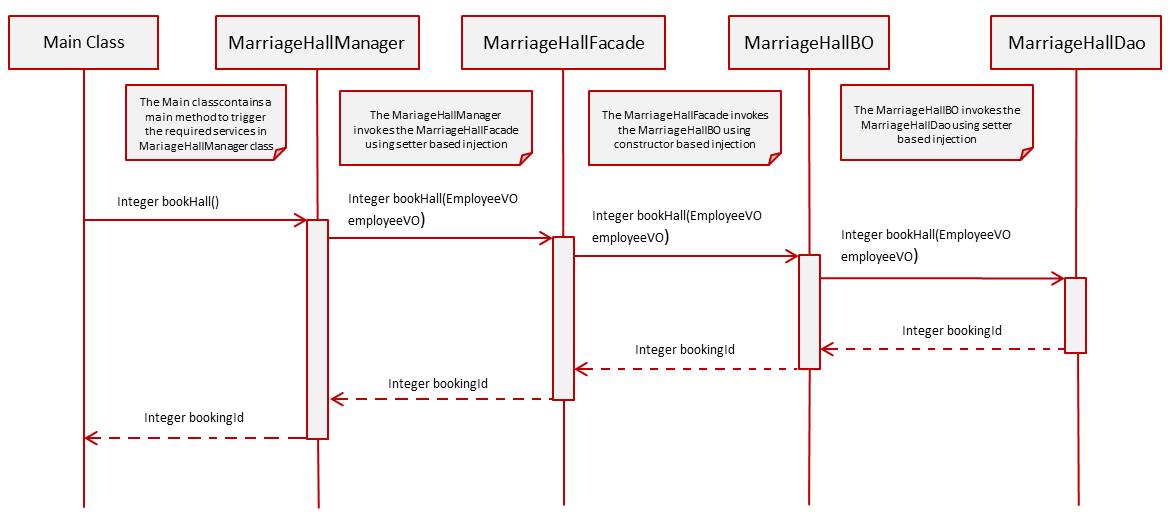


**Output**





**Sequence Diagram**

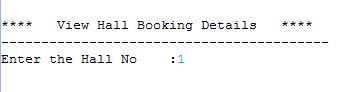


**View Hall Booking Details:**

When the user selects Option 2, viewHallBookingDetails () method of Tester is invoked.

The Marriage Hall admin will put in the hallNo.

The ViewHallBookingDetails menu in the ViewHallBookingDetails () method of Tester class looks like below



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

**In the Tester Class,**

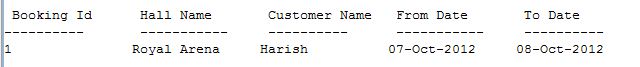
Invoke viewHallBookingDetails (Integer hallNo) from Tester to MarriageHallManager following the Control flow mentioned above. This method returns a list of type DetailsVO objects.

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:** While displaying, to display fromDate and toDate in proper Date Format (dd-MMM-yyyy),

Use DateUtility.getStringFromDate (Date date) to convert registrationDate from Date to String]

**OUTPUT**



**Technical Specifications:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component Name | Method Name | Input | Output | Exception |
| MarriageHallManager | viewHallBookingDetails() | Integer hallNo | List<DetailsVO> detailVoList | NoRecordException  This Exception has to be caught and thrown back to Tester class. |
| MarriageHallFacade | viewHallBookingDetails() | Integer hallNo | List<DetailsVO> detailVoList | NoRecordException  This Exception has to be caught and thrown back to the MarriageHallManager class. |
| MarriageHallBO | viewHallBookingDetails() | Integer hallNo | List<DetailsVO> detailVoList | NoRecordException  This exception has to be caught and thrown back to MarriageHallFacade class. |
| MarriageHallDAO interface | viewHallBookingDetails() | Integer hallNo | List<DetailsVO> detailVoList |  |
| MarriageHallDAOImpl | viewHallBookingDetails() | Integer hallNo | List<DetailsVO> detailVoList |  |
| hall\_config.xml | Contains all the xml configurations related to Service 2(View Hall Booking Details) |  |  |  |

**Business Rules & Validations:**

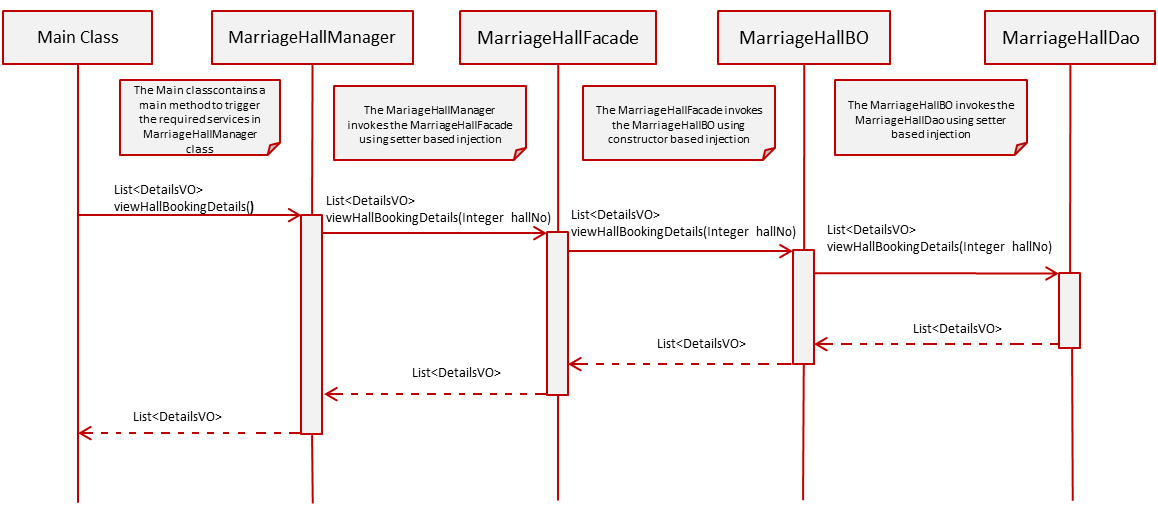
In the MarriageHallBO class check for the following business rules:

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

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

**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 the name and method signature.)

**Sequence Diagram**



**Solution:**



**Evaluation Rubrics**

|  |  |
| --- | --- |
| **Parameters** | **Weightage** |
| Completeness |  |
| Accuracy |  |
| Clarity of understanding |  |
| Presentation |  |
| Total |  |

**Summary of this Case Study:**

You have just learnt:

* Spring Framework and IOC Container
* Spring Dependency Injection
* Spring JDBC Framework
* Spring Transaction Management
* ResourceBundleMessageSource