|  |
| --- |
| cog logo difference tag  coverpageimage2  ­    Employee Management System |
|  |

Contents

[Employee Management System 2](#_Toc483475253)

[1. Important Instructions 2](#_Toc483475254)

[2. Introduction 2](#_Toc483475255)

[3. Software Checklist 2](#_Toc483475256)

[4. Skeleton Code for Development 2](#_Toc483475257)

[5. Backend Database Setup 3](#_Toc483475258)

[6. Technical Requirements 4](#_Toc483475259)

[I. Scenario – Update Employee Location details 4](#_Toc483475260)

[II. Design Rules 5](#_Toc483475261)

[III. Sequence Diagram: 6](#_Toc483475262)

[7. Solution 6](#_Toc483475263)

# Employee Management System

## Important Instructions

A skeleton archive file with the project structure, necessary XML’s, properties file and library files will be provided to the associate prior to the assessment. Associate should use this zip file to develop the solution. Associate should follow the below mentioned Do’s and Don’ts while developing the solution.

**Do’s for the developing the Solution.**

* Import the provided skeleton code slice project into the workspace and start the development.
* You can add any methods as necessary.
* Ensure that your uploaded code has no compilation errors.
* The requirement should be implemented based on the design considerations (Specified in the Limitations & Constraints section in the Case Study) specified.
* You can use either annotations or XML for implementing the solution.

**Don’ts for the developing the Solution.**

* Do not change the imported archive file name.
* Do not add or remove Library files.
* Do not change the package and file names given in the case study.
* Do not hard code the input parameters inside the program.

## Introduction

A start- up organization maintains the records of all the employees and the location details of the organization in the database. Only for few facilities the associate can get transfer. There should be an application that provides the details to the associate to let them know whether they can transfer the location which they are requesting for. As a software consultant/developer, build an application using Spring Core (version 4.x) to ease this process.

## Software Checklist

* Eclipse SDE 7.0
* Java SE 1.8
* MySQL 5.6.x
* Spring 4.x jars

## Skeleton Code for Development

Unzip the below content and place it in your hard disk

**skeleton**

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.

## Backend Database Setup

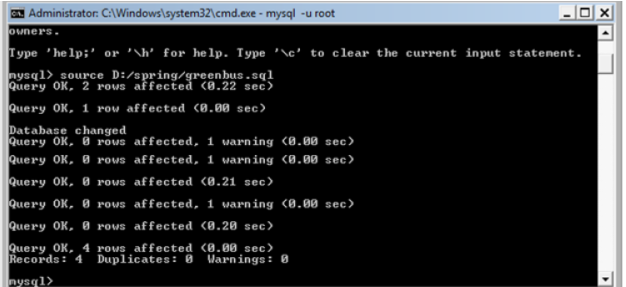
1. Unzip the below content and place it in your hard disk

**sql**

1. Open MySQL Command Line and run the script using the command:

**Example: source D:/Javaca/PlaceOrder.sql** (after unzipping the sql file please run this command by giving the location where the .sql file is placed)

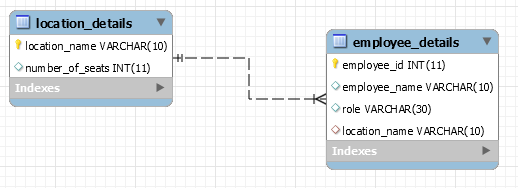
Queries should execute without any errors. Below screenshot is an example.



**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.

## Technical Requirements

Refer to below ER Diagram



**Requirement 1:**

### Scenario – Update Employee Location details

* An Employee in an organization requests for the transfer to other location.
* In order to initiate the transfer, first the Employee must check the transfer availability of the given location using property file.
* Only if the transfer is eligible for the requested location, transfer will be provided.
* If the location is eligible for transfer, update the Employee details with the requested transfer location. In the meantime, the respective number of seats for the transfer location should be decreased only if it has sufficient number of available seats

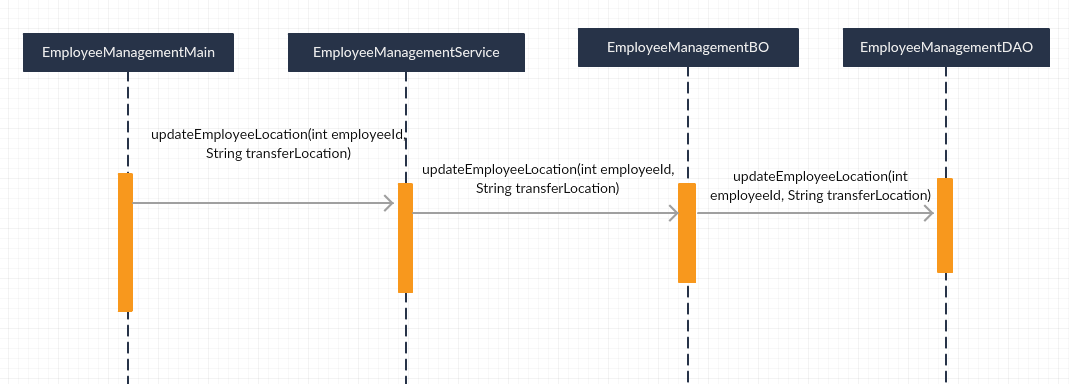
### 

|  |  |  |  |
| --- | --- | --- | --- |
| Component Name | Method Name | Input | Output |
| EmployeeManagementService.java | updateEmployeeLocation | employeeId, transferLocation | void |
| EmployeeManagementBO.java | getTransferEligiblityStatus | transferLocation | String |
| EmployeeManagementBO.java | updateEmployeeLocation | employeeId, transferLocation | void |
| EmployeeManagementDAO.java | updateEmployeeLocation | employeeId, transferLocation | void |

### Design Rules

1. You can configure using either XML or annotations to implementation the solution
2. EmployeeManagementBO should be autowired with EmployeeManagementService
3. EmployeeManagementDAO should be autowired with EmployeeManagementBO
4. MessageSource should be autowired in EmployeeManagementBO
5. Jdbctemplate should be autowired in EmployeeManagementDAO using setter based dependency injection.
6. Transaction management should be enabled in EmployeeManagementBO in the business method ‘updateEmployeeLocationDetails’
7. In order to update the location details into the database, first retrieve the transfer availability status for the requested location in the BO layer using the ResourceBundle in the message ‘getTransferAvailabilityStatus()’.
8. If the requested transfer location is not available in the property file, throw the application exception with the error message as “Invalid Location”
9. Only if the returned status is “Y’ , then proceed to DAO to update the location of the Employee
10. If the returned status is “N”, then throw the business exception “EmployeeManagementException” with the error message as “Transfer is not Eligible for this location”
11. After updating the location details of the employee, the number of seats in the respective location should be decreased by 1.
12. If the number of seats is zero, then throw the application exception with the error message as “NO seats available. Raise a request for the new seat”.
13. Only if the number of seats are available then the update for number of seats should happen

### Sequence Diagram:



## Solution

SOLUTION