Employee Referral Program

Duration: 3 Hrs.

Problem Statement:

ABC organization needs a portal for the employees that helps them to refer for the skills that are on demand. Create a web portal to fulfill their requirements. To fulfill the requirements, the following functionalities need to be implemented.

1. Make a referral
2. Show referral

Database:

Create table **skillset** and **employee** in **MYSQL** database by following the description given below and insert records.

Table: skillset

Column Name	Datatype	Description
skillId	int(10)	primary key
skill	varchar(20)	
level	varchar(5)	
bonus	double	

Table: employee

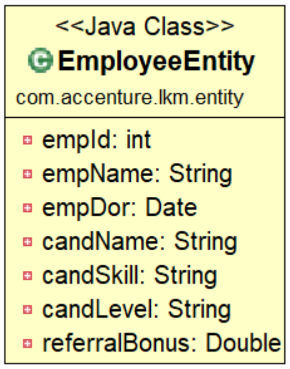
Column Name	Datatype	Description
empId	Int	Primary key, auto
empName	varchar(20)	
empDor	Date	
candName	varchar(20)	
candSkill	varchar(20)	
candLevel	varchar(20)	
referralBonus	Double	

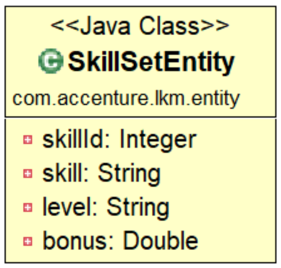
skillset Records

SkillId	skill	level	Bonus
11	J2EE	10	10000
12	J2EE	9	15000
13	SAP ABAP	9	16000
14	SAP ABAP	10	12000
15	Sales Force	10	11000
16	Sales Force	8	20000

ENTITY BEAN CLASS:

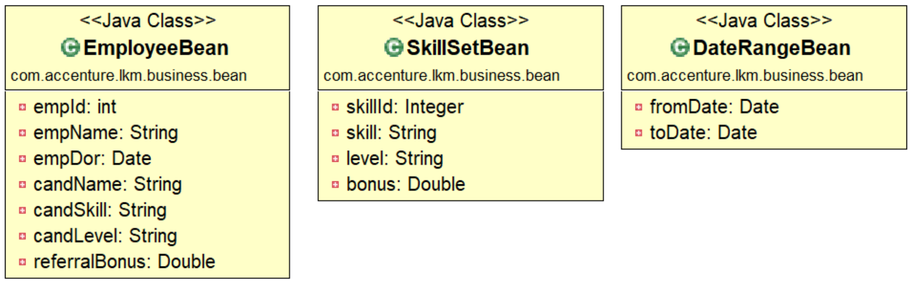
Create entity bean classes **EmployeeEntity , SkillSetEntity** and map to table **employee** and **skillset** and auto generate the employee Id value**.** Generate getter and setter methods for all properties.





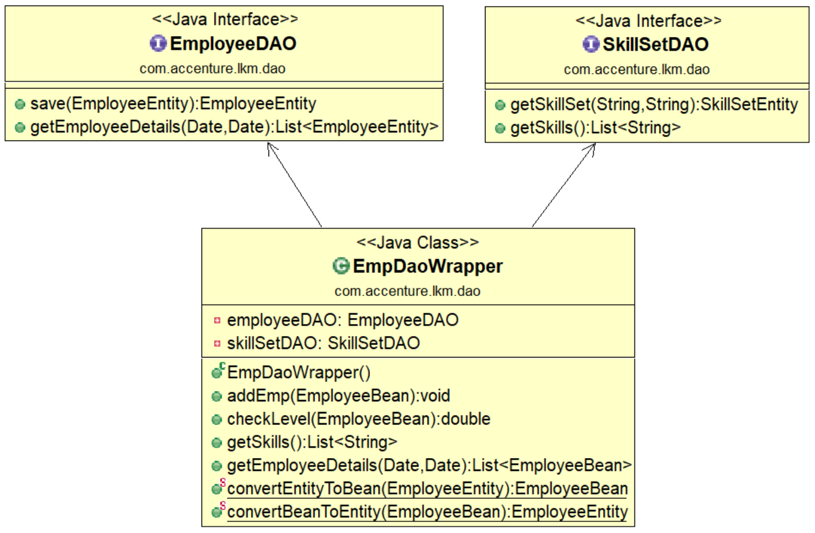
BEAN CLASSES:

Create bean class **EmployeeBean, SkillSetBean** and **DateRangeBean** as per the class diagram given below. Generate getter and setter methods for all properties. These bean classes can be used as data transfer objects.



DAO LAYER:

Create two interfaces **EmployeeDAO**, **SkillSetDAO** and **EmpDaoWrapper** class as per the class diagram given below. Provide the required annotations in interfaces to enable **Spring JPA Data** and **Spring JPA.** Provide the required annotations to the properties in wrapper class for autowiring.



<<Java Interface>>
EmployeeDAO
com.accenture.lkm.dao
save(EmployeeEntity):EmployeeEntity
getEmployeeDetails(Date,Date):List<EmployeeEntity>
<<Java Interface>>
SkillSetDAO
com.accenture.lkm.dao
getSkillSet(String,String):SkillSetEntity
getSkills():List<String>
<<Java Class>>
EmpDaoWrapper
com.accenture.lkm.dao
employeeDAO: EmployeeDAO
skillSetDAO: SkillSetDAO
EmpDaoWrapper()
addEmp(EmployeeBean):void
checkLevel(EmployeeBean):double
getSkills():List<String>
getEmployeeDetails(Date,Date):List<EmployeeBean>
convertEntityToBean(EmployeeEntity):EmployeeBean
convertBeanToEntity(EmployeeBean):EmployeeEntity

Note: Queries should be defined in orm.xml.

Following methods need to be defined in interfaces **EmployeeDAO**, **SkillSetDAO**

EmployeeDAO interface :

save():

This is repository method to insert the medicine order details in the respective table.

getEmployeeDetails():

This should invoke the proper query to fetch the employee referral details in the given date range.

SkillSetDAO interface:

getSkillSet():

This should invoke the proper query to fetch the skillset details for the respective skillLevel and skill.

getSkill():

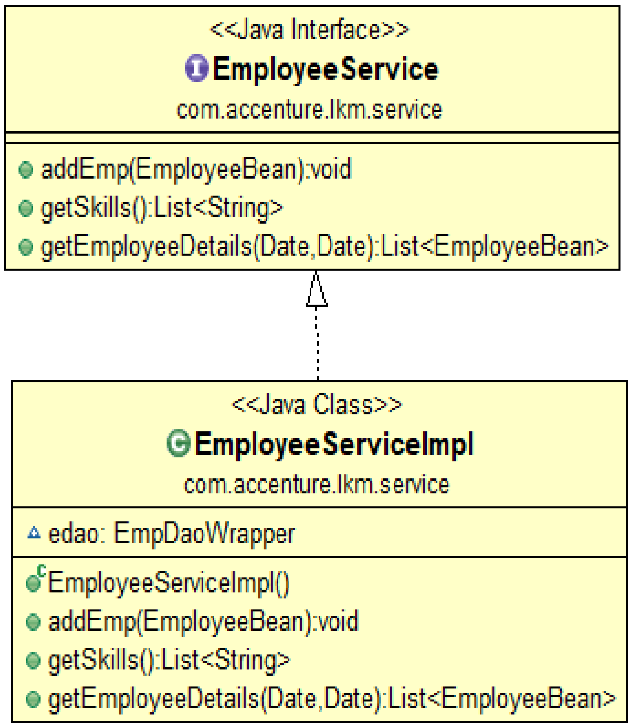
This should invoke the proper query to fetch the distinct skills from skillset.

Following methods need to be implemented in **EmpDaoWrapper.** Using **Spring JPA and Spring JPA Data**
[note:- while writing JPA, use the injected **EntityManager** to perform database operations.]

checkLevel()	It verifies the match of skill and level with the database data using **Spring JPA Data.** If match found, then return the bonus.
addEmp()	It should use BeanUtils to convert bean to entity. Invoke checkLevel() method by passing empBean to find the skill - level match. If there is no match it must throw **Skill-Level mismatch Exception**. If there is a match, get the bonus and assign to entity object and persist the same using **Spring JPA Data**
getSkills()	It fetches the skill details from SkillSetEntity using **Spring JPA Data** and store the skills into a list object that helps to populate skill drop down list.
getEmployeeDetails()	Find the employee details who referred the candidates between given fromDate and toDate using **SpringJPA** and convert them into a list of bean objects and return the same.

SERVICE LAYER:

Create the **EmpService** interface and **EmpServiceImpl** class as per the class diagram given below. These methods are used for business logic implementation and invoking the respective **EmpDAOWrapper** methods. Autowire edao using proper annotation.



<<Java Interface>>
EmployeeService
com.accenture.lkm.service
addEmp(EmployeeBean):void
getSkills():List<String>
getEmployeeDetails(Date,Date):List<EmployeeBean>
<<Java Class>>
EmployeeServiceImpl
com.accenture.lkm.service
edao: EmpDaoWrapper
EmployeeServiceImpl()
addEmp(EmployeeBean):void
getSkills():List<String>
getEmployeeDetails(Date,Date):List<EmployeeBean>

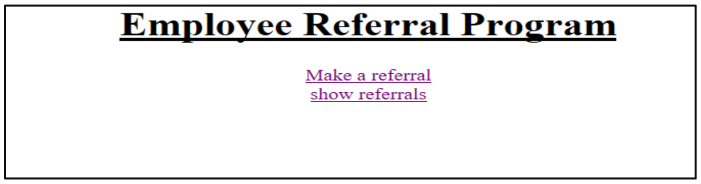
addEmp()	It should invoke **addEmp()** of **EmpDaoWrapper** to save the details of employee.
getSkills()	It should invoke **getSkills()** of **EmpDaoWrapper**

	to get the list of skills that can be populated in the dropdown list.
getEmployeeDetails()	It should invoke **getEmployeeDetails()** of **EmpDaoWrapper** to get all employee details who referred the candidates in the given date range

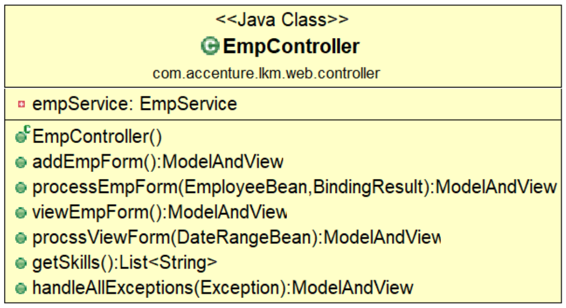
<u>PRESENTATION LAYER</u>

<u>index.jsp</u>

Create **index.jsp** as shown below. This should be the **home page** of the application



Create **EmpController** class as per the class diagram given below

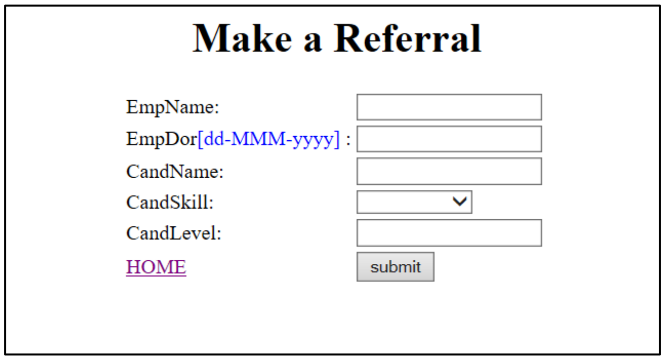

<<Java Class>>
EmpController
com.accenture.lkm.web.controller
empService: EmpService
EmpController()
addEmpForm():ModelAndView
processEmpForm(EmployeeBean,BindingResult):ModelAndView
viewEmpForm():ModelAndView
procssViewForm(DateRangeBean):ModelAndView
getSkills():List<String>
handleAllExceptions(Exception):ModelAndView

addEmpForm()	It should create and populate **ModelAndView** object with **AddEmpForm** logical view name and **EmployeeBean** instance as a model.
processEmpForm()	It should be mapped to appropriate URL. It should retrieve **EmployeeBean** submitted by the view. It should invoke addEmp() of service class by passing the **EmployeeBean** retrieved above. It should create and populate **ModelAndView** object with **success** logical view name and appropriate message as a model [Refer screenshot for appropriate message] Validation is done for the EmployeeBean object, if the mandatory fields are not filled, then it returns **ModelAndView** object with **AddEmpForm** logical view name to display the validation error message.
getSkillList()	It invokes getSkills() method of service class and returns the list of skills, so that the list can be prepopulated in the dropdown list.
viewEmpForm()	It should create and populate **ModelAndView** object with **DateRangeBean** object **EmployeeReport** as a logical view name
processViewForm()	It should get **DateRangeBean** object from **EmployeeReport** form and invoke **getEmployeeDetails()** of service class that returns list of **EmployeeBean.** If the size of returned list is >0 then pass the list to **CandidateDetails** otherwise send the error message and display the same page **[EmployeeReport]** Refer the screenshot for appropriate message]

handleAllException()	It should catch the **exception** object and add appropriate exception message with **ModelAndView** object and **GeneralizedExceptionHandlerPage** logical view name to display the exception message.

AddEmpForm.jsp

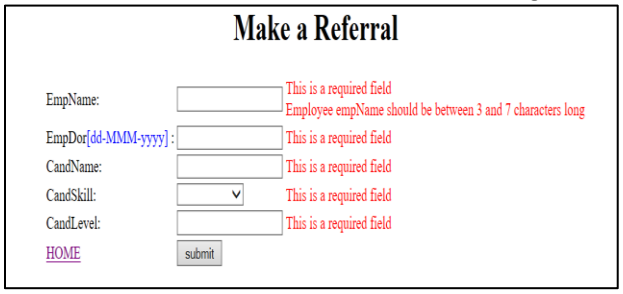
When **Make a referral** link is clicked from **index.jsp** then **AddEmpForm.jsp** should be displayed as shown below:



Make a Referral

EmpName:	
EmpDor[dd-MMM-yyyy] :	
CandName:	
CandSkill:	
CandLevel:	
HOME	submit

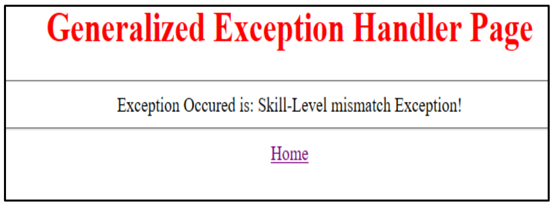
- Bind all the fields to appropriate data members of **EmpBean**
- **candSkill** should populate dynamically. Use **getSkillList()** of controller class.
- If the mandatory fields are not filled, then the validation error message should as shown below



Make a Referral

EmpName:		This is a required field Employee empName should be between 3 and 7 characters long
EmpDor[dd-MMM-yyyy] :		This is a required field
CandName:		This is a required field
CandSkill:		This is a required field
CandLevel:		This is a required field
HOME	submit	

- If there is a mismatch in the skill and level, then display the following exception message **GeneralizedExceptionHandlerPage.jsp**

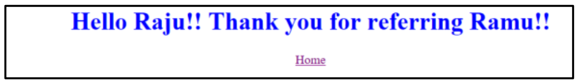


Generalized Exception Handler Page

Exception Occured is: Skill-Level mismatch Exception!

Home

- On providing the required valid details and clicking the **submit** button, **success.jsp** should be displayed with the success message as shown below:

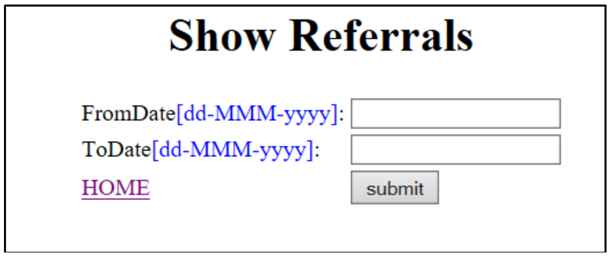


Hello Raju!! Thank you for referring Ramu!!

Home

EmployeeReport.jsp

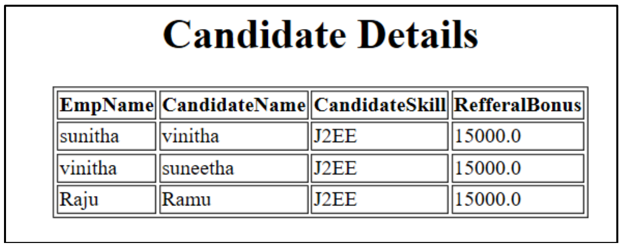
- When **show referral** link is clicked from **index.jsp** then **EmployeeReport.jsp** should be displayed as shown below:



Show Referrals

FromDate[dd-MMM-yyyy]:	
ToDate[dd-MMM-yyyy]:	
HOME	submit

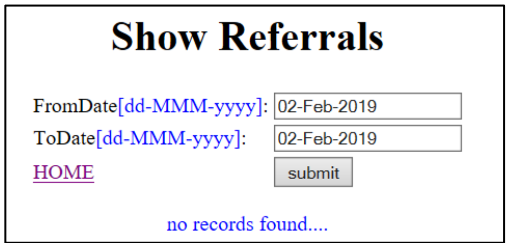
- once user enters the details and click on **submit** button, the employee referral details should be displayed in **CandidateDetails.jsp** the format given below:



Candidate Details

EmpName	CandidateName	CandidateSkill	RefferalBonus
sunitha	vinitha	J2EE	15000.0
vinitha	suneetha	J2EE	15000.0
Raju	Ramu	J2EE	15000.0

- If no records are not available in the given date range then display the following message in **EmployeeReport.jsp**



Show Referrals

FromDate[dd-MMM-yyyy]:	02-Feb-2019
ToDate[dd-MMM-yyyy]:	02-Feb-2019
HOME	submit

no records found....