

Problem Statement:

Dan Pizza store needs a portal to manage the customer orders. Create a web portal to fulfil their requirements. To fulfil the requirements, the following functionalities need to be implemented.

1. Save Order
2. Order Report

Database:

Create tables Pizza and Pizza_Order in **MYSQL** database by following the description given below and insert records.

Table: Pizza

Column Name	Data Type	Description
pizzald	integer	primary key
pizzaName	varchar(20)	
price	double	

Pizza Records:

pizzald	pizzaName	price
1001	XYZVegS	200
1002	XYZVegM	400
1003	XYZVegL	600
1004	XYZNonVegS	400

Table: Pizza_Order

Column Name	Data Type	Description
orderId	integer	Primary key, Auto Increment
customerName	varchar(30)	
contactNumber	varchar(10)	
pizzald	integer	Foreign key refers to Pizza tale
numberOfPiecesOrdered	integer	
bill	double	

Pizza_Order Records

orderId	customer Name	contact Number	pizzald	numberOfPieces Ordered	bill
5001	Peter	1234567890	1001	1	200
5002	Thomas	6574893012	1001	2	400
5003	Decken	1243568790	1003	1	600
5004	Jenifer	1029384756	1004	2	800

ENTITY BEAN CLASSES:

Create entity bean classes **PizzaEntity** and **PizzaOrderEntity** and map to tables **Pizza** and **Pizza_Order** tables respectively. Generate getter and setter methods for all properties.

com.accenture.lkm.entity.PizzaEntity
▪ pizzald: Integer
▪ pizzaName: String
▪ price: Double

com.accenture.lkm.entity.PizzaOrderEntity
▪ orderId: Integer
▪ pizzald: Integer
▪ customerName: String
▪ contactNumber: String
▪ bill: Double
▪ numberOfPiecesOrdered: Integer

BEAN CLASSES:

Create bean class **PizzaBean**, **PizzaOrderBean** and **BillRangeBean** 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.

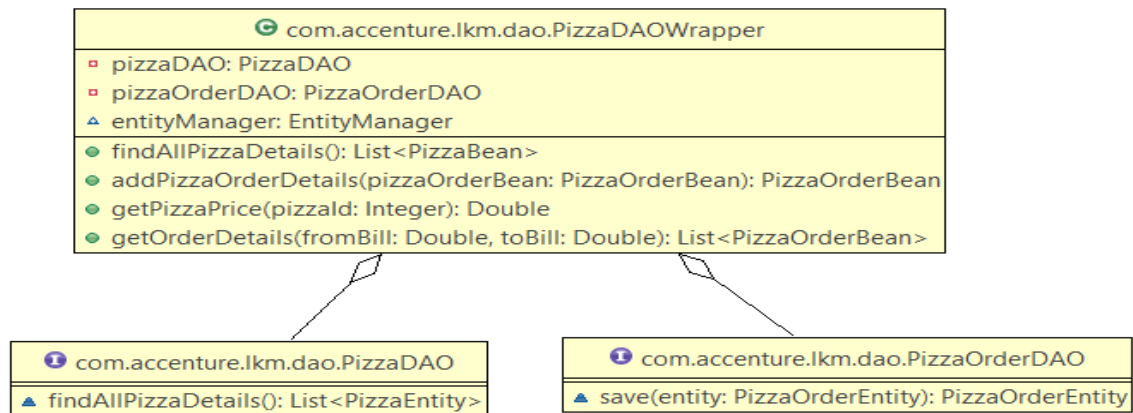
com.accenture.lkm.business.bean.PizzaBean
<ul style="list-style-type: none"> ▪ pizzald: Integer ▪ pizzaName: String ▪ price: Double

com.accenture.lkm.business.bean.PizzaOrderBean
<ul style="list-style-type: none"> ▪ orderId: Integer ▪ pizzald: Integer ▪ customerName: String ▪ contactNumber: String ▪ bill: Double ▪ numberOfPiecesOrdered: Integer

com.accenture.lkm.business.bean.BillRangeBean
<ul style="list-style-type: none"> ▪ fromPrice: Double ▪ toPrice: Double

DAO LAYER:

Create **PizzaDAOWrapper** class and interface **PizzaDAO** and **PizzaOrderDAO** as per the class diagram given below. Provide the required annotations in interfaces to enable **Spring JPA Data**



Following method needs to be created in **PizzaDAO** using **orm.xml** and **Spring JPA Data**

findPizzaDetails():	It should map to the appropriate query in orm.xml to fetch all the pizza details [Note: Query should be written in orm.xml appropriately]
----------------------------	---

Following method needs to be created in **PizzaOrderDAO** using **orm.xml** and **Spring JPA Data**

save():	This is repository method to insert the pizza order details
----------------	---

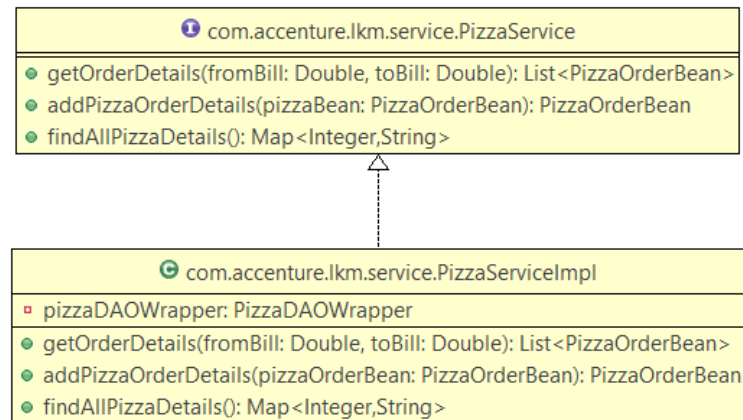
Following methods need to be created in **PizzaDAOWrapper**. [**Note:** While writing JPA, use the injected **EntityManager** to perform the database operations]

findAllPizzaDetails()	It should invoke findAllPizzaDetails() method of PizzaDAO to get the details of pizza. It should use BeanUtils to convert entity to bean. It should return the details
addPizzaOrderDetails()	It should use BeanUtils to convert bean to entity. It should invoke save() method of PizzaOrderDAO to insert the order details and return the obtained object

getPizzaPrice()	Use JPA to implement this requirement.It should retrieve the price of the given pizza and return
getOrderDetails()	Use JPA to implement this requirement.It should fetch all order details between the given bill amount and return

SERVICE LAYER:

Create the **PizzaService** interface and **PizzaServiceImpl** class as per the class diagram given below. These methods are used for business logic implementation and invoking the respective **PizzaDAOWrapper** methods.



getOrderDetails()	It should invoke getOrderDetails() of PizzaDAOWrapper to get the details of pizza and return. If there is no order placed, then throw an exception with appropriate error message
addPizzaOrderDetails()	It should invoke getPizzaPrice() of PizzaDAOWrapper to get the price of the selected pizza. It should calculate the bill using unit price and number of pieces ordered, set the bill to bean object and invoke addPizzaOrderDetails() of PizzaDAOWrapper and return the obtained object
findAllPizzaDetails()	It should invoke findAllPizzaDetails() of PizzaDAOWrapper to get the all pizza details. It should populate a map with pizza Id as key and Pizza name as value

PRESENTATION LAYER (Save Order)

index.jsp

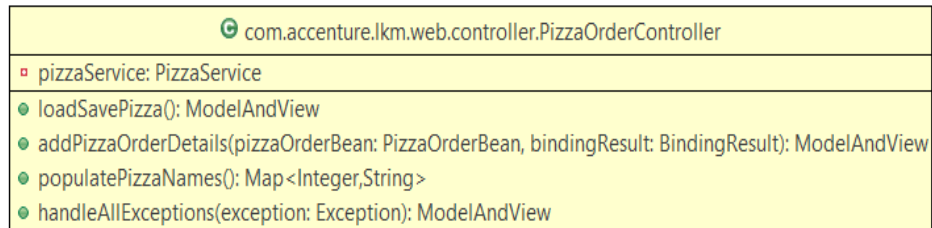
- Create **index.jsp** as shown below. This should be the **home page** of the application

Dan Pizza's

[Save Order](#)

[Order Report](#)

- Create **PizzaOrderController** class as per the class diagram given below



loadSavePizza()	It should create and populate ModelAndView object with PizzaOrder as logical view name and PizzaOrderBean instance as Model
saveCutomer()	It should be mapped to appropriate URL. It should retrieve PizzaOrderBean submitted by the view. It should check whether any form validation failed or not. If failed, populate ModelAndView object with PizzaOrder logical view name to display the validation error message. On successful form validation, it should invoke the addPizzaOrderDetails() of service class by passing the PizzaOrderBean retrieved above . get the orderId and bill from the returned object. It should create and populate ModelAndView object with PizzaOrderSuccess as logical view name and appropriate message as model [Refer screen shot for appropriate message]
populatePizzaNames()	This method is used to populate the model attribute. It should invoke findAllPizzaDetails() of PizzaService to get pizza details and return the map
handleAllException()	This is a common exception handler. It should create and populate ModelAndView object with GeneralizedExceptionHandlerPage as logical view name and exception message as model

PizzaOrder.jsp

- When **Save Order** link is clicked from **index.jsp** then **PizzaOrder.jsp** should be displayed as shown below:

The screenshot shows a web form titled "Add Pizza Details". It contains four input fields: "Customer Name", "Customer Contact", "Pizza Name" (a dropdown menu with "--Select--" selected), and "Quantity". Below the fields is an "Order" button and a "Home" link.

- Bind all the fields to appropriate data members of **PizzaOrderBean**
- Pizza Name is populated dynamically. Use **populatePizzaName()** of controller class
- Validation: use standard validation to make sure that all fields are mandatory, minimum quantity ordered should be 1, contact number should have 10 digits, customer name should have 3 to 30 characters

- On providing the required valid details and clicking the **Order** button, **addPizzaOrderDetails()** of **PizzaOrderController** should be invoked and in case of successful execution **PizzaOrderSuccess.jsp** should be displayed as shown below:

PizzaOrderSuccess.jsp

Add Pizza Order Details Success

Hi: Mark, your order is placed with orderId: 5008, Bill to be paid is: 400.0

[Home](#)

- If any input field validation failed, then display the error message as shown below

Add Pizza Details

Customer Name	ab
Customer Contact	123
Pizza Name	--Select--
Quantity	

[Home](#)

All Errors

Customer Name should have length between 3 and 30

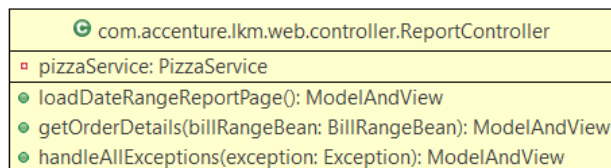
Please select a valid Pizza

Contact Number should have length 10

Quantity should not be empty

PRESENTATION LAYER (Order Report)

- Create **ReportController** class as per the class diagram given below



loadDateRangeReportPage()	It should create and populate ModelAndView object with OrderReport as logical view name and BillRangeBean instance as Model
getOrderDetails	It should be mapped to appropriate URL. It should retrieve BillRangeBean submitted by the view. It should invoke the getOrderDetails() of service class by passing the BillRangeBean . It should create and populate ModelAndView object with OrderReport as logical view name and List<PizzaOrderBean> as Model

handleAllException()	This is a common exception handler. It should create and populate ModelAndView object with GeneralizedExceptionHandlerPage as logical view name and exception message as model

OrderReport.jsp

- When **Save Order** link is clicked from **index.jsp** then **PizzaOrder.jsp** should be displayed as shown below:

Order with in Price Range Report

[Home](#)

- Once user click on **Fetch Details** button, the order details should be displayed in the format given below:

Order with in Price Range Report

Order Id	Customer Name	PizzaId	Bill	Quantity
5001	Peter	1001	INR.200.00	1
5002	Thomas	1001	INR.400.00	2
5003	Decken	1003	INR.600.00	1
5007	Mark	1001	INR.400.00	2
5008	Mark	1001	INR.400.00	2

[Home](#)

- If there is any exception, it should be displayed in common error page **GeneralizedExceptionHandlerPage.jsp** as shown below:

Generalized Exception Handler Page

Exception Occurred is: No records were found for the entered Bill Range

[Home](#)