## Instructions:

You will build a web-based online pizza ordering system using Java EE. This system will allow customers to order pizzas online for pickup or for home delivery. You are given a **PizzaDB** to start with. You may need to alter this database as the project progresses.

For this project, you may work in **groups of 2**.

## Business Case:

Pizza restaurants are a multi-billion dollar industry and they are literally across the globe. As the number of internet users are increasing, many companies are introducing Online Pizza Ordering Systems for taking orders from customers. These systems not only improve customer experience, but also ease the workload for restaurant staff.

Ordering a pizza by phone seems easy to do but sometimes there is miscommunication. Since there is no visual menu shown during the phone call, restaurant employees have to repeat many things again and again to customers. This is a time consuming process that can irritate customers and take a lot of time for restaurant staff.

It would be much more comfortable for customers to order their pizza online. It would be hassle free for users as they can select the pizza and toppings they want and make payment for it.

In summary, online pizza ordering will improve the service provided to the restaurant's customers. From a business point of view, it gives you an edge over competitors.

## Actors and Goals

### Customer

**Build Pizza** –Customer will be able to order a custom pizza. Customer will be able to choose exactly the toppings they want on their pizza. Customer will be able to choose the crust type and size (Small, medium, large) of the pizza they want. Note: an order being built exists only in memory, only when the order is complete is it saved to the database.

**Order confirmation** – Customer will see all the details of their order before finalizing the order. This confirmation will help customers to check the items ordered with their prices. The customer can choose to pay online, or pay on pickup.

**Know Delivery / Pickup Time** – Customer will see the time by which the order will be delivered. For pick-up orders, the customer can choose a later time when they will pick up their order.

**Customer Details** – The Customer table has 8 `non-null` fields, but no password field ☹. If you like, you can add a password, and then add a Customer Login screen.

### Employee

**Login** – Employee must be able to login securely to an "Administration" area of the website that is only available to authorized users.

**Manage Toppings** – Employee must be able to add/remove available toppings from the "Administration" area of the website. (sometimes the kitchen runs short, and a topping is temporarily unavailable)

**Manage Orders** – Employee must be able to view orders from the "Administration" area of the website. Employee can choose to view "pending orders" or "filled orders". Employee must able to change an order status from "pending" to "filled".

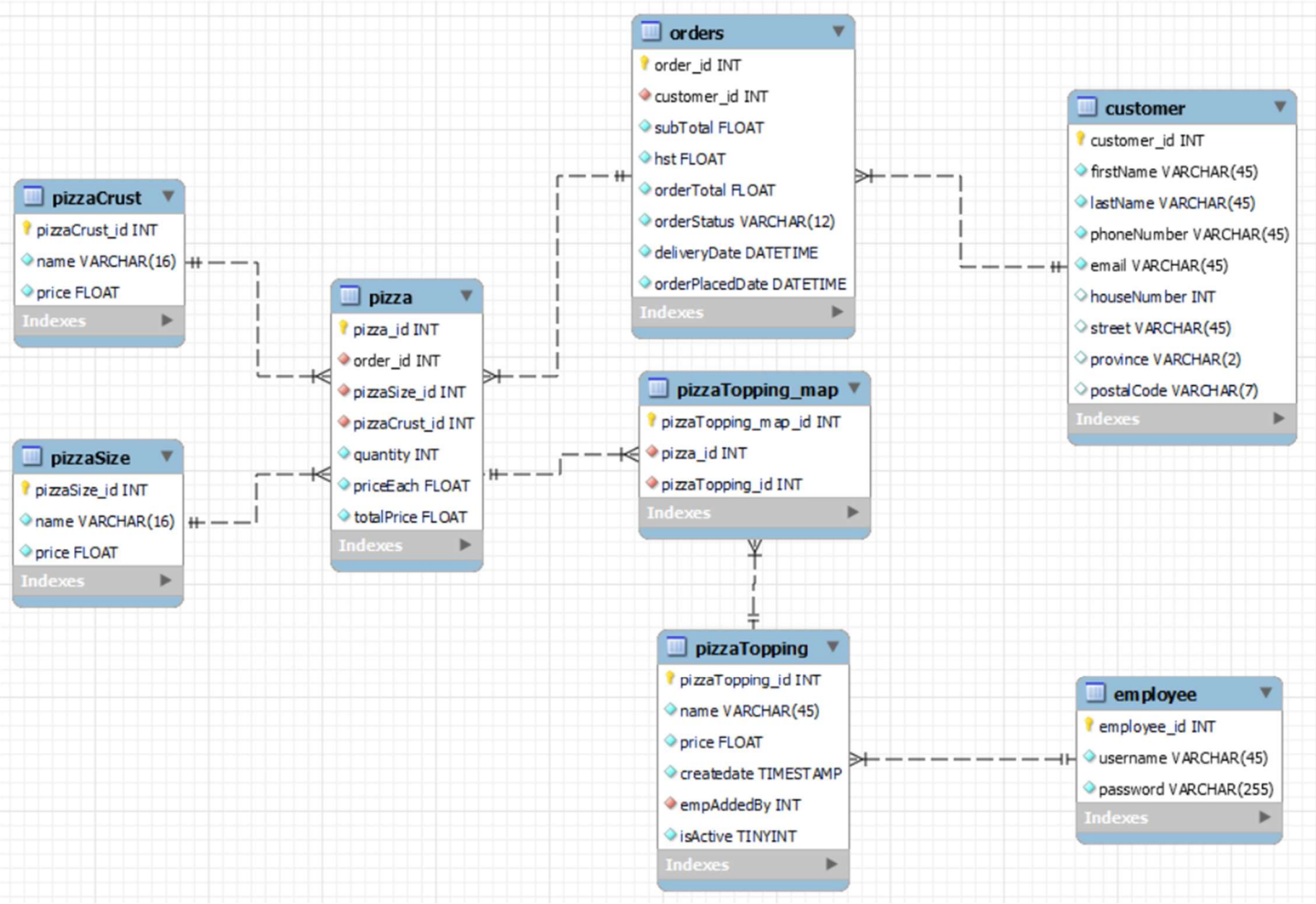
**Supplementary Specifications**

1. Upon visiting the website, the customer will be presented with a menu of available pizza sizes and crust types, and a list of all the toppings that are currently available in the system from which they can choose. All toppings will be shown with their prices.
2. Customer must be able to customize their pizza and add it to their cart.
3. Customer must be able to select a quantity of a particular pizza (1-5) when adding it to their cart.
4. Customer must be able to order more than 1 pizza at a time.
5. System must include 15% sales tax on all orders. The System will estimate the preparation time and delivery time. The customer can choose a later time, if desired.
6. After selecting the items to be ordered, customer must select order type (home delivery or pickup) and provide personal details like name, phone number, address, email etc.
7. Customer must have option to pay online or cash on delivery (COD).
8. After ordering, the customer will see what time the pizza will be ready/delivered.

**Technical Specifications**

1. You must use Java classes for all backend DB operations
2. For the frontend webpages, you can use HTML/JavaScript/REST, or JSP / JSF pages. You can even do some pages as JSP and some as JSF, if you want to
3. The "Cart" functionality can be done with Session variables.
   *(There is no need to save temporary data in the database)*
4. For online payments, simply redirect the user to http://PayPal.com as a placeholder
5. Proper validation must be performed at each stage and exceptions must be caught to make sure the users never see any obscure error messages
6. You must organize your code using proper 3-tier architecture with a Presentation Layer, Business Layer and Data Layer
7. Don't spend too much time on graphics and CSS but all website pages must be attractive, controls lining up etc. You can "borrow" images from the internet to jazz up your website a bit
8. Assume all customers will have Canadian addresses and phone numbers
9. An employee can change available toppings **at any time** – therefore, the list of available toppings should be refreshed periodically, or refreshed using checking a system-wide flag.

## ER Diagram




## Database Tables Summary

- **pizzaCrust, pizzaSize, employee** – these tables are Read Only
- **pizzaTopping** – this table needs full CRUD support, for employees only
  *Note: toppings are not physically deleted – to delete a topping, just set isActive=0*
- **customer, orders** –Create, Update ability needed here (Delete not required)
- **pizza, pizzaTopping_map** – only Create (and Read) ability is needed here

## Deliverables

1. You must submit two progress reports outlining what you have been working, and what you plan to work on in the next week, and any impediments that are slowing you down or preventing you from making progress. These progress reports must be submitted through BrightSpace and are due on the following dates:
   - Progress Report #1: Sunday, December 1st
   - Progress Report #2: Sunday, December 8th

2. **The completed Project is Due: Wednesday, December 11th**
   You must submit your Project as a single ZIP file.
   If you made any changes in the database structure, the modified SQL file must be included.
   If you are using a REST API, a complete & detailed specification document must be included.
   The ZIP must contain an SQL database dump of your MySQL PizzaDB database.
   **Note**: No extensions will be granted except under truly exceptional circumstances

3. **Presentations will be scheduled for Thursday & Friday**

## Grading

5% Status Report 1

5% Status Report 1

80% Completed Application

10% Presentation