# Online Ordering Pizza Application

## Overview

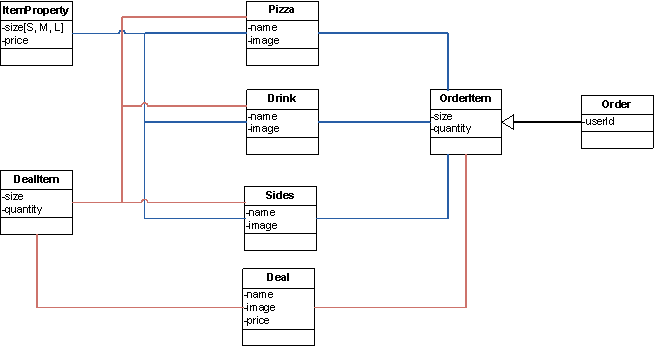
As a new developer in R&D, you will implement a simple “Pizza Ordering” web application. You will develop an API using Spring Boot and a client application that will consume the API using VueJs (Quasar).   
This application must be completed in **3 weeks** and you will present and demonstrate your project to the R&D leadership team, as well as other guests.

*Be as innovative as possible to add, modify & beautify functionalities within the time frame while fulfilling all the basic requirements listed.*

Part of your first probation evaluation will be based on this project, so take it seriously and good luck!

## Data Model

### Model



### Sample Data

|  |  |
| --- | --- |
| **Pizza** | |
| **name** | **Image** |
| Vegetarian Pizza | images/pizza/veg\_pizza.jpg |
| Cheese Pizza | images/pizza/cheese\_pizza.jpg |
| Chicken Pizza | images/pizza/chicken\_pizza.jpg |

|  |  |
| --- | --- |
| **Sides** | |
| **name** | **Image** |
| Fries | images/sides/fries.jpg |
| Wings | images/sides/wings.jpg |
| Cookie | images/sides/coookie.jpg |

|  |  |  |
| --- | --- | --- |
| **deal** | | |
| **Name** | **image** | **Price** |
| Family Deal | images/deal/fam\_deal.jpg | 25 |
| Movie Night Deal | images/deal/movie\_deal.jpg | 21 |
| Sunday Deal | images/deal/sun\_deal.jpg | 15 |

### Assumptions

* Price for the deal is defined on the deal itself hence no computation needed for calculating the price
* Size of the Item is standardized to S, M and L. However, you can create separate Properties for each item to have customized size for each kind of item
* Image URL can be stored in the database and images are stored as files in the web location

## Dependencies

To develop the API, you will use the latest version of Spring Boot. You will also use JPA and Mysql database for the persistence side of things.

## Business Requirements

1. Search Items to order
   1. Create searches for the items listed below
      1. Pizzas (Eg, Veg Pizza, Chicken Pizza, etc.,)
      2. Drinks (Eg. Coke, Fanta, etc)
      3. Sides (Eg. Fries, Wings, etc.,)
      4. Deals (Eg. Family Deal (1Large Pizza, 2 small Drinks, 1 medium fries))
   2. Both Suggestions and search results page must be configured to display the results
   3. The results should display the name with the image
   4. Using the context menu in the search the user should be able to add the item to the current order
2. Ordering
   1. The user should be able to add items to the current order, for that you need to enable POST requests to save the orders that the end-user will make.
   2. Once the item is added to the current order it is saved. When the user logs out and login back, he should be able to see the saved item in the current order
   3. The user should be able to save the order for future ordering
3. My orders – **Component**
   1. My orders list all the saved orders of the user
   2. The number of orders which can be saved should be **configurable**
   3. Saving orders more than that should delete the oldest order automatically and save the new one
   4. User should be able to delete any stored order from this component
4. Current Order – **Component**
   1. Items added to the current order from search should be displayed here
   2. User can remove the items directly from this component
   3. User has the option to choose the size and qty of the item
   4. Price for each item is displayed according to the chosen size of the item
   5. Total Price is calculated by this formula

Sum of the price + Provincial Tax + Federal Tax + Delivery Charges

* 1. All tax percentages and delivery charges must to **configurable**
  2. The order can be saved only when the minimum order price is reached
  3. The minimum order price must be configured as well
  4. The current is order is always saved and the user can logout/login and still his order is persisted

1. For sibling components to communicate with each other and to retrieve/send data from/to your API, you will need to create a service.
2. Menu
   1. New menu Item is created for “My orders” which opens the My Orders component to the view stack
   2. New menu Item is created for “Current Order” which opens the Current Order component to the view stack
3. You will need an exception handler for your API in order to properly deal with eventual exceptions.

## Technical Requirements

1. Follow strict Java coding and naming conventions
2. Make sure the application works in Chrome, Firefox and IE
3. Make sure the application is responsive
4. Code should adhere to testability, reusability, scalability and maintainability
5. Concentrate both on UI and UX for better user experience

## Innovation

The requirements listed here are the least we expect from the project but that does not contain your innovation or creativity. You can go beyond the requirements and develop some cool features on your project and to achieve the same you can add or modify the data model as needed. At least one unique feature other than the requirement is nice to have and impress the Leadership team.

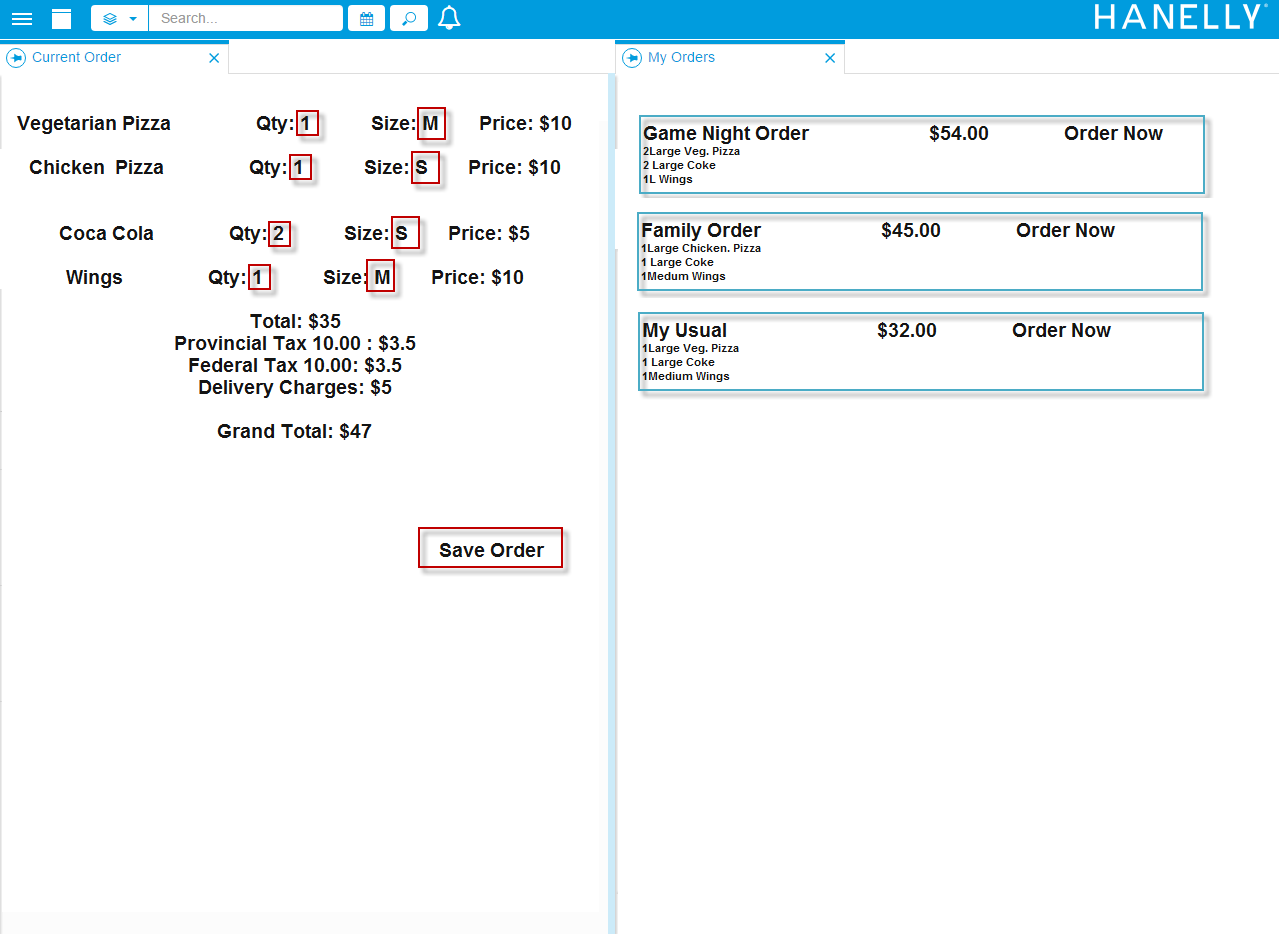
*Note: Please complete the basic requirements first before jumping on new features. As we have strict timeline it is better to complete the project and add additional features.*

## Test and Document the Pizza API

Add tests for the Pizza Api classes (Coverage).

## Rough Mockup

*Note: This screenshot is purposefully made very basic so not to direct you along any UI design. Please treat this as a basic outline for your component. The UI has to be much nicer, beautiful and functional.*



## Helpful Tips for the Demo

* Introduce yourself so that the review team gets a sense of your experience and skills.
* Show line numbers in Eclipse and increase the font size as you will be projecting to make the review easier
* Verify that your project looks the way you expect before the demo – using the same projector/room where your demo is to take place
* Let the team know about known bugs and issues so that the focus can be on your work, and not on your bugs.
* Schedule a 1-on-1 demo with your team’s lead developer before your main demo at least a day or two