The Pizza Time! pizza customization app is a demo project consisting of a UI, REST API, and heavily encapsulated Java backend. As well as a full suite of associated tests.

I began by sketching an idea of how I wanted the UI to look. Then I began building each element from the most basic components. I used Enums for Pizza Topping type (Cheese, Pepperoni, Pineapples, etc), amounts (Light, Medium, Extra), and Pizza Size (Small, Medium, Large), as these values would be strict unless additional values were added in the future.

The most basic element is a Pizza Topping, with its topping and amount values. Then a Pizza consists of a size, and a list of toppings. A default pizza with no attributes is set to a Medium, Cheese only pizza. The highest level is the Pizza Order, which contains a list of pizzas, as well as additional data such as order number, the date the order is started, and a total cost of all the pizzas and their toppings.

For the size of this project, I opted for an in-memory solution for storing the prices of the toppings and the pizza sizes. There is also a multiplier to the price for topping amount.

I created a very robust REST API for accessing all the components. A pizza order can be created with default or custom pizzas. Pizzas can be added or removed from the pizza order. The pizza size can be updated. Individual toppings can be added and removed, or toppings can be updated as a list.

All the logic for the REST API controller takes place in the PizzaOrderService class. This handles all functions associated with the controller, as well as the methods for calculating the cost of the order and the cost per pizza after changes. This value is updated in the PizzaOrder object after every change.

There is a test for all functions in the model and in the main service. As well as a simple test for the controller and the application context.

The UI is built on AngularJS, HTML, Bootstrap, and CSS. On load the API is called to create a new PizzaOrder with a default pizza. Every reload starts a new default order. Toggles are used to add or remove toppings from the pizza. If added, a drop down appears for specifying the topping amount. These changes are immediately reflected in the “Order Form” on the right. Total Cost is also automatically recalculated and added. The size dropdown will update the size attribute of the pizza, and the new size and price are immediately displayed as well.

There is a pizza “image” displayed by css on the page. Adding and removing cheese or pepperoni will update this display.

For the scope of this project, I opted to leave out adding, editing, and deleting additional pizzas and the cost per pizza, but the API does support it. I also forewent updating all toppings in the image, and only chose two to mock out the functionality. I had also started to add a “Check Out” feature with some additional functionality.

In a full application, I would use a database as opposed to an in-memory solution for the pricing and a “Admin Only” API for accessing those changes.

Let me know if you have any problems with the application, I only tested on my setup so it may not work or run as expected on other setups.

Looking forward to discussing this design more!