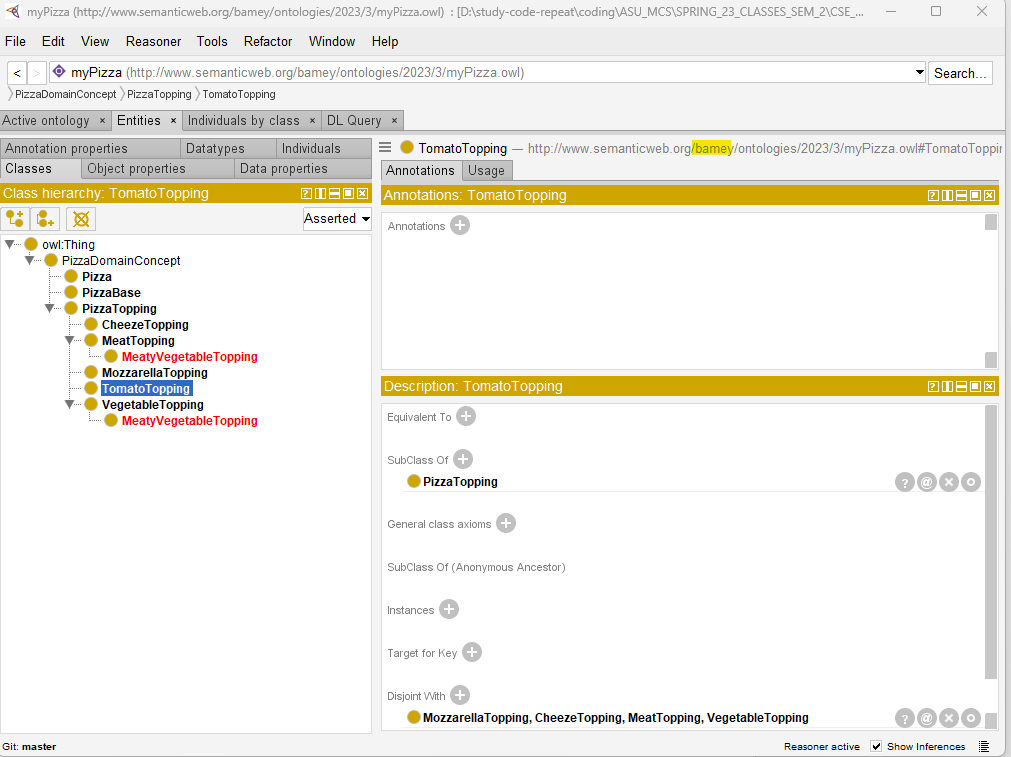
**Programming Assignment 3: Building an OWL Ontology**

Amey Bhilegaonkar

**Q1. Do any of your classes come out as inconsistent? Explain why and describe a way to resolve the inconsistency?**

**Ans:** Yes, 2 of the classes came out as inconsistent when we added disjoints.When we introduced the topping MeatyVegetableTopping and assigned it to both MeatTopping and VegetableTopping categories, we observed inconsistency in the classes. This is because the same topping cannot be assigned to two different topping categories.



**Explanation** **of inconsistency:**

We were trying to make things a bit more organized by creating a new category called "MeatyVegetableTopping" that would be a subcategory of both "MeatTopping" and "VegetableTopping". But then, we ran into a bit of a problem when we tried to make "MeatTopping" and "VegetableTopping" totally separate from each other. You see, if we made "MeatyVegetableTopping" totally separate from both "MeatTopping" and "VegetableTopping", then we'd also be making it separate from itself! And that just doesn't make sense. So basically, we've got an inconsistency on our hands. It's not ideal, but it's something we'll need to address and figure out a solution for.

Graphical user interface, text, application, email

Description automatically generated

**Resolution:** To fix the problem we encountered with the MeatyVegetableTopping category, one possible solution would be to adjust the way we've organized our classes. Instead of trying to make MeatTopping and VegetableTopping completely separate from each other, we could make them disjoint only from other types of toppings.

This means that while MeatTopping and VegetableTopping would still be separate categories, they wouldn't be completely separate from each other. This would allow us to include the MeatyVegetableTopping category as a subcategory of both MeatTopping and VegetableTopping, without causing any logical inconsistencies.

Graphical user interface

Description automatically generated

**Descriptions:**

I added following properties:

1. hasCrunch: Its corresponding class has 3 choices, soft, medium and hard crunch
2. hasCrust: Its corresponding choices are Lemon and Garlic Crust
3. hasNationality: Its corresponding choices are Indian, American, Italian
4. hasSpice: It choices are it has, spice toppings.

I added following Toppings:

1. Indian Toppings
2. Spicy Toppings

I added following Bases:

1. White Base
2. Wheat Base

I added following Pizzas with properties:

1. PunePizza:
   1. hasNationality: Indian
   2. hasCrust: Lemon
   3. hasSpice: Spicy Topping, Indian Topping
   4. hasCrunch: Medium Crunch
2. PaneerPizza:
   1. hasNationality: Indian
   2. hasCrust: Garlic
   3. hasSpice: Aloo Topping, Indian Topping
   4. hasCrunch: Hard Crunch

**Q6. First query of your own choice.**

**Ans:** NamedPizza and hasNationality some Indian and hasTopping some IndianTopping

**Q6-1. Explanation of the First query.**

**Ans:** This query finds all the NamedPizza types which have Indian Nationality, and only those which have IndianTopping.

**Q6-2. Screenshot of the First query.**

**Ans: Graphical user interface, text, application

Description automatically generated**

**Q7. Second query of your own choice.**

**Ans:** hasCrust some LemonCrust

**Q7-1. Explanation of the Second query.**

**Ans:** This query finds all the hasCrust types which have LemonCrust

**Q7-2. Screenshot of the Second query.**

**Ans: Graphical user interface, application

Description automatically generated**