

# Model Driven Engineering

## Assignment 4

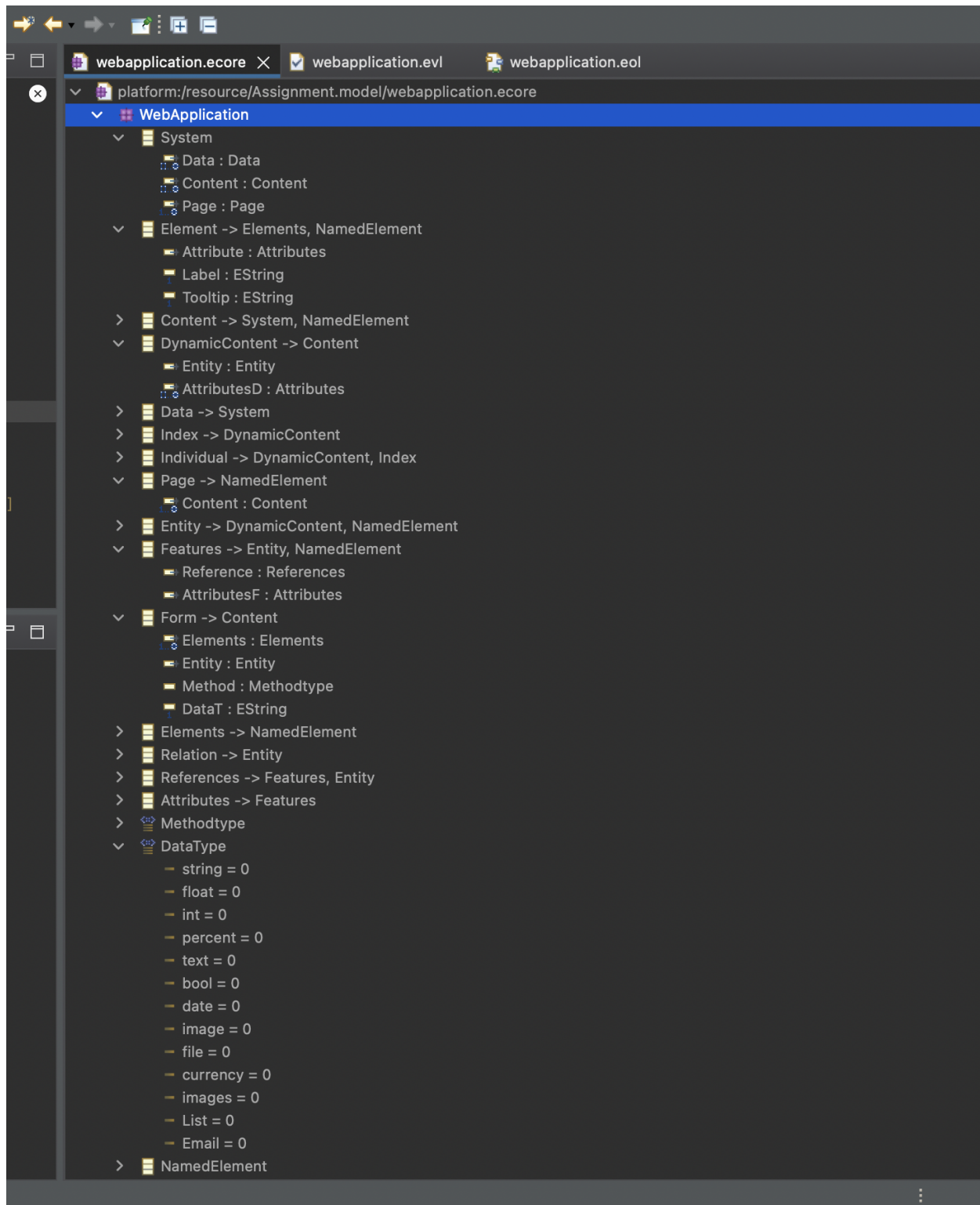
*By*

***Team Chicken Restaurant:***

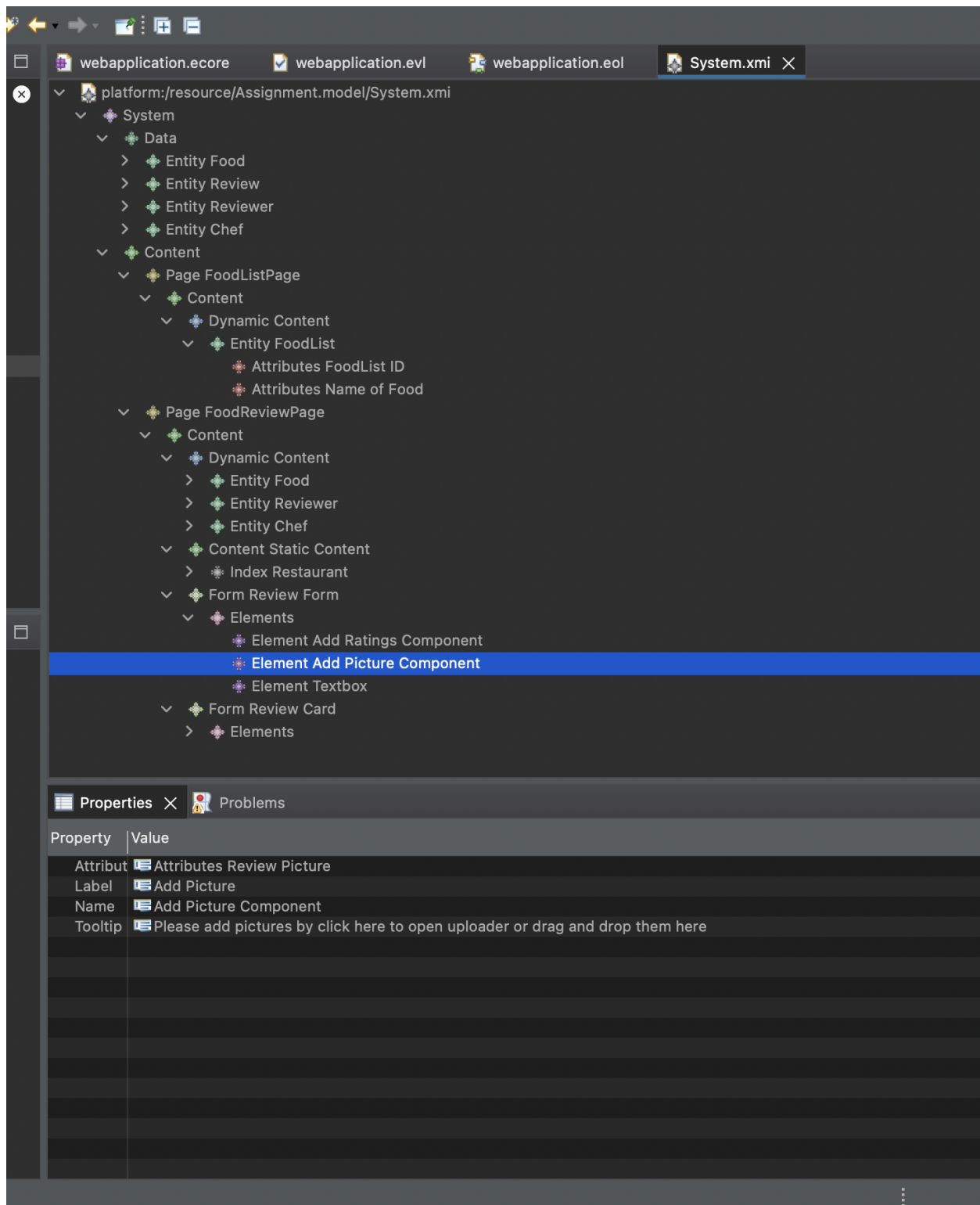
Cindy Aprilia  
Korawit Rupanya  
Mercy Bamiduro

1)

Based on our previous projects developed on MPS, we defined 16 metaclasses. Each metaclass contains at least one attribute or reference as demonstrated in the screenshot below. Also, the metamodel was properly defined and contains inheritance, containment, enumeration types, attributes, data types, etc.

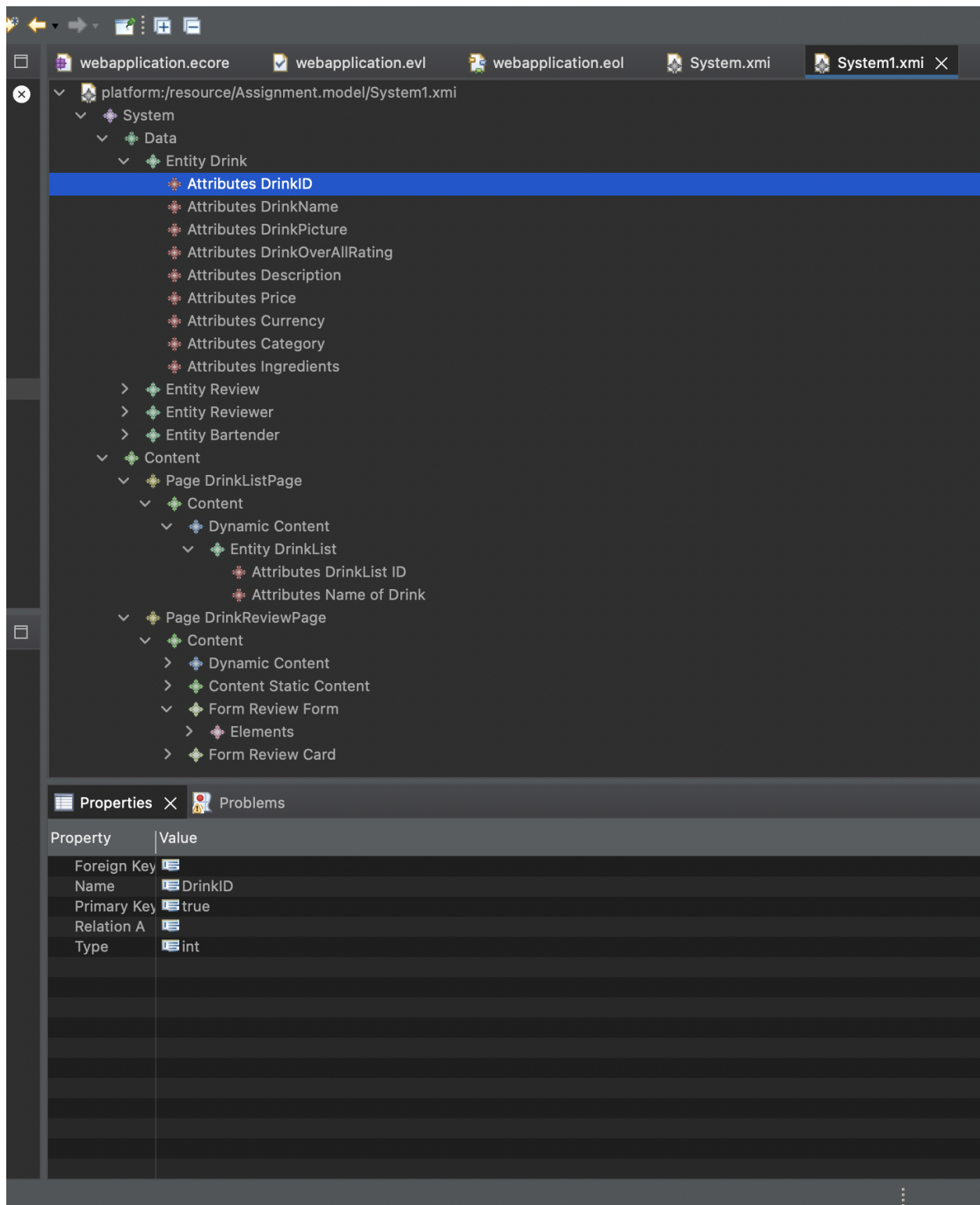


Our metamodel was further instantiated by the first concrete instances named “**System.xml**” with each concept represented in the above metaclasses, well instantiated in the model as seen in the image below;



2b)

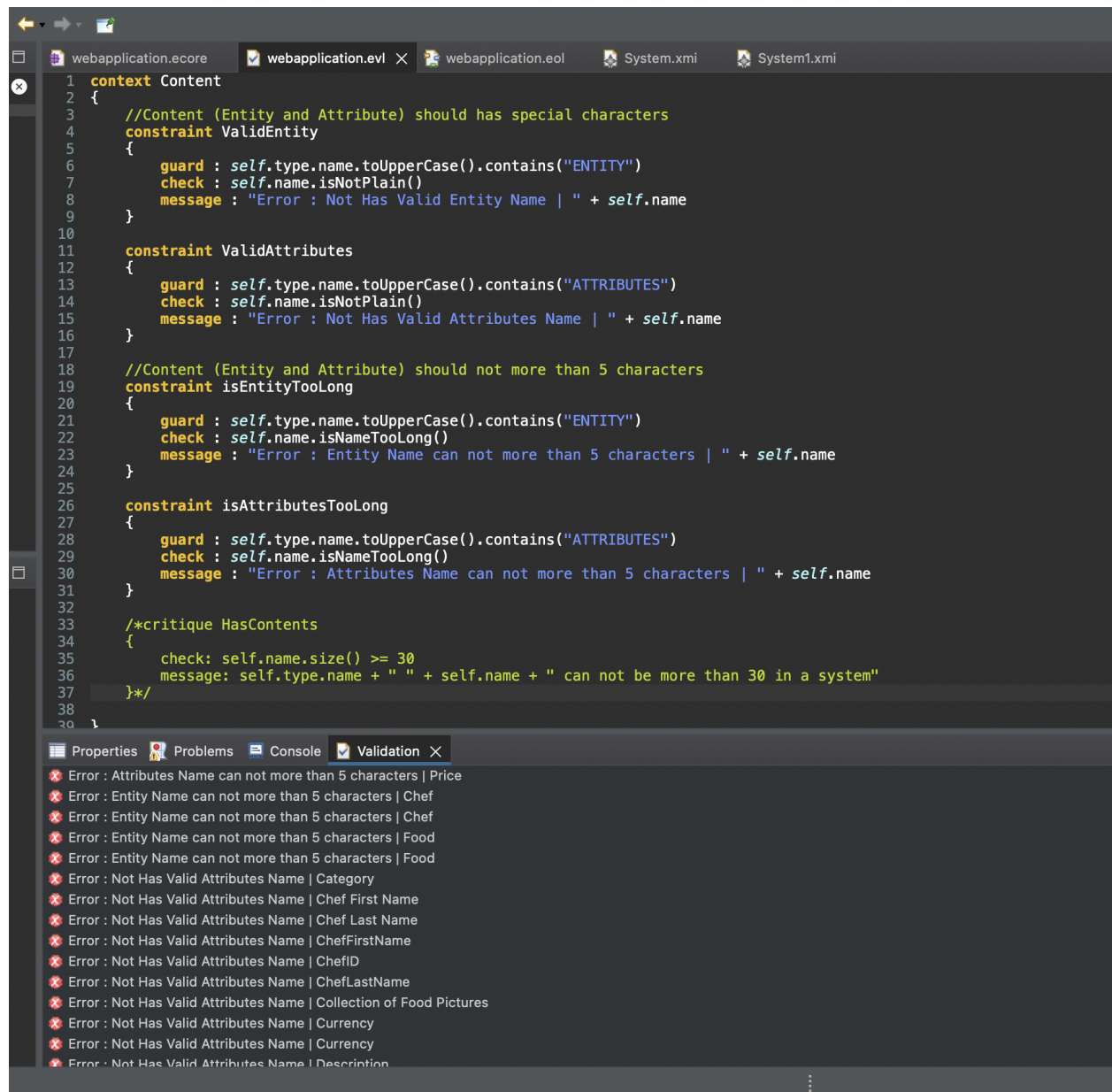
Additionally, the metamodel was further instantiated by the second concrete instance named “**System1.xmi**” with each concept also represented in the above metaclasses, well instantiated in the model too as seen in the image below;



3)

We have defined the following constraints and critique for both models as demonstrated in the image below and available in the video recording attached also.

- To check if the Entity has special characters.
- To check if the Attributes have special characters.
- To check if the Entity name has more than 5 characters.
- To check if the Attributes name has more than 5 characters.
- To check if the Content is more than 30 in a system.



```
1 context Content
2 {
3     //Content (Entity and Attribute) should has special characters
4     constraint ValidEntity
5     {
6         guard : self.type.name.toUpperCase().contains("ENTITY")
7         check : self.name.isNotPlain()
8         message : "Error : Not Has Valid Entity Name | " + self.name
9     }
10
11     constraint ValidAttributes
12     {
13         guard : self.type.name.toUpperCase().contains("ATTRIBUTES")
14         check : self.name.isNotPlain()
15         message : "Error : Not Has Valid Attributes Name | " + self.name
16     }
17
18     //Content (Entity and Attribute) should not more than 5 characters
19     constraint isEntityTooLong
20     {
21         guard : self.type.name.toUpperCase().contains("ENTITY")
22         check : self.name.isNameTooLong()
23         message : "Error : Entity Name can not more than 5 characters | " + self.name
24     }
25
26     constraint isAttributesTooLong
27     {
28         guard : self.type.name.toUpperCase().contains("ATTRIBUTES")
29         check : self.name.isNameTooLong()
30         message : "Error : Attributes Name can not more than 5 characters | " + self.name
31     }
32
33     /*critique HasContents
34     {
35         check: self.name.size() >= 30
36         message: self.type.name + " " + self.name + " can not be more than 30 in a system"
37     }*/
38 }
```

Properties Problems Console Validation X

- ✖ Error : Attributes Name can not more than 5 characters | Price
- ✖ Error : Entity Name can not more than 5 characters | Chef
- ✖ Error : Entity Name can not more than 5 characters | Chef
- ✖ Error : Entity Name can not more than 5 characters | Food
- ✖ Error : Entity Name can not more than 5 characters | Food
- ✖ Error : Not Has Valid Attributes Name | Category
- ✖ Error : Not Has Valid Attributes Name | Chef First Name
- ✖ Error : Not Has Valid Attributes Name | Chef Last Name
- ✖ Error : Not Has Valid Attributes Name | ChefFirstName
- ✖ Error : Not Has Valid Attributes Name | ChefID
- ✖ Error : Not Has Valid Attributes Name | ChefLastName
- ✖ Error : Not Has Valid Attributes Name | Collection of Food Pictures
- ✖ Error : Not Has Valid Attributes Name | Currency
- ✖ Error : Not Has Valid Attributes Name | Currency
- ✖ Error : Not Has Valid Attributes Name | Description

webapplication.ecorewebapplication.evlwebapplication.eolSystem.xmiSystem1.xmi

```
1 context Content
2 {
3     //Content (Entity and Attribute) should has special characters
4     constraint ValidEntity
5     {
6         guard : self.type.name.toUpperCase().contains("ENTITY")
7         check : self.name.isNotPlain()
8         message : "Error : Not Has Valid Entity Name | " + self.name
9     }
10
11     constraint ValidAttributes
12     {
13         guard : self.type.name.toUpperCase().contains("ATTRIBUTES")
14         check : self.name.isNotPlain()
15         message : "Error : Not Has Valid Attributes Name | " + self.name
16     }
17
18     //Content (Entity and Attribute) should not more than 5 characters
19     constraint isEntityTooLong
20     {
21         guard : self.type.name.toUpperCase().contains("ENTITY")
22         check : self.name.isNameTooLong()
23         message : "Error : Entity Name can not more than 5 characters | " + self.name
24     }
25
26     constraint isAttributesTooLong
27     {
28         guard : self.type.name.toUpperCase().contains("ATTRIBUTES")
29         check : self.name.isNameTooLong()
30         message : "Error : Attributes Name can not more than 5 characters | " + self.name
31     }
32
33     critique HasContents|
34     {
35         check: self.name.size() >= 30
36         message: self.type.name + " " + self.name + " can not be more than 30 in a system"
37     }
38
39 }
```

PropertiesProblemsConsoleValidation

Attributes Category can not be more than 30 in a system

Attributes Chef First Name can not be more than 30 in a system

Attributes Chef Last Name can not be more than 30 in a system

Attributes ChefFirstName can not be more than 30 in a system

Attributes ChefID can not be more than 30 in a system

Attributes ChefLastName can not be more than 30 in a system

Attributes Collection of Food Pictures can not be more than 30 in a system

Attributes Currency can not be more than 30 in a system

Attributes Currency can not be more than 30 in a system

Attributes Description can not be more than 30 in a system

Attributes Food Description can not be more than 30 in a system

Attributes Food Ratings can not be more than 30 in a system

Attributes FoodID can not be more than 30 in a system

Attributes FoodID can not be more than 30 in a system

Attributes FoodID can not be more than 30 in a system

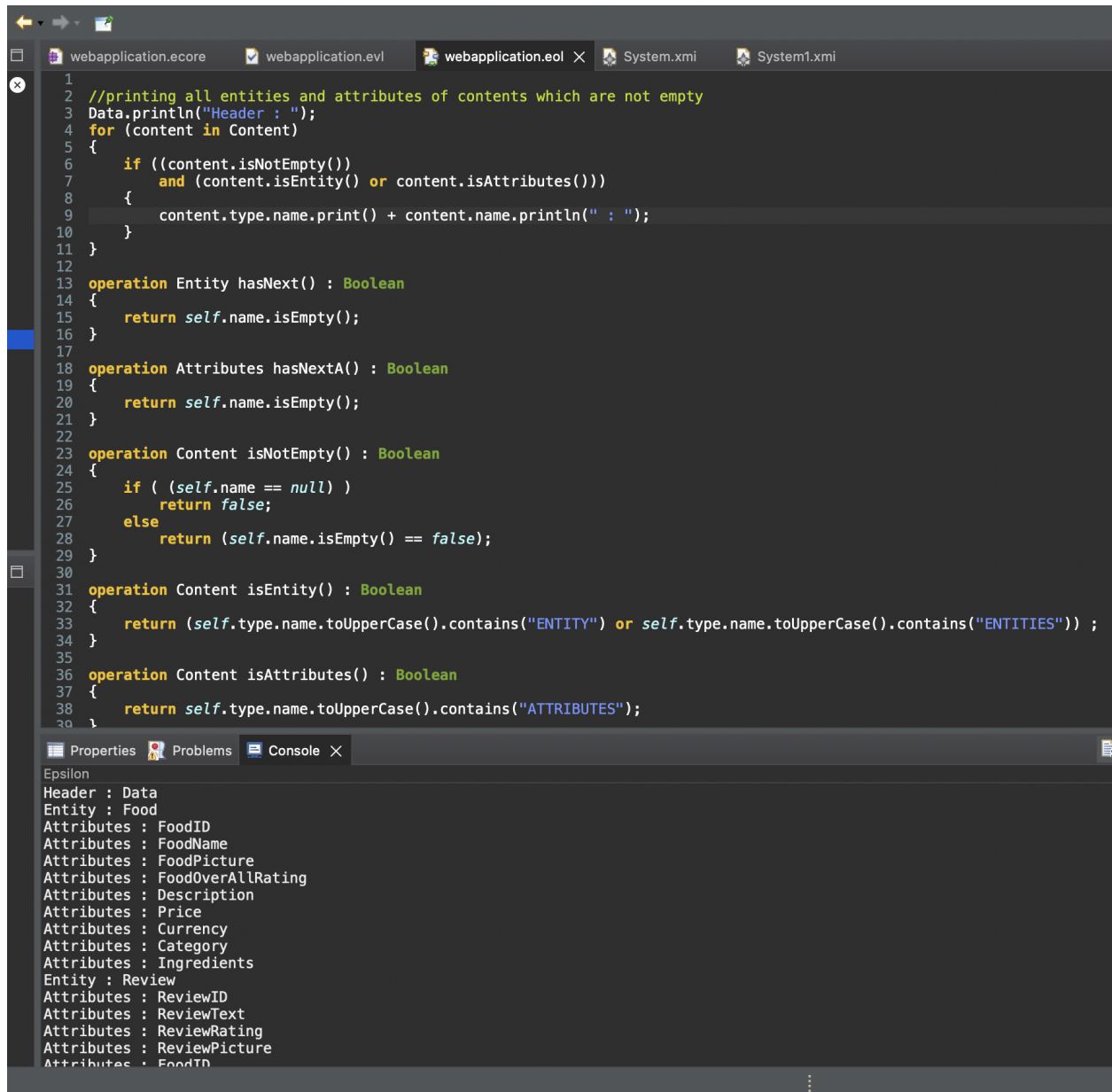
WritableSmart Insert33 : 25 : 1019



3b)

We have also defined the following operations as seen in the image below and in the video recording.

- Printing all entities of contents that are not empty.
- Printing all attributes of contents that are not empty.



```
1
2 //printing all entities and attributes of contents which are not empty
3 Data.println("Header : ");
4 for (content in Content)
5 {
6     if ((content.isNotEmpty())
7         and (content.isEntity() or content.isAttributes()))
8     {
9         content.type.name.print() + content.name.println(" : ");
10    }
11 }
12
13 operation Entity hasNext() : Boolean
14 {
15     return self.name.isEmpty();
16 }
17
18 operation Attributes hasNextA() : Boolean
19 {
20     return self.name.isEmpty();
21 }
22
23 operation Content isEmpty() : Boolean
24 {
25     if (self.name == null)
26         return false;
27     else
28         return (self.name.isEmpty() == false);
29 }
30
31 operation Content isEntity() : Boolean
32 {
33     return (self.type.name.toUpperCase().contains("ENTITY") or self.type.name.toUpperCase().contains("ENTITIES"));
34 }
35
36 operation Content isAttributes() : Boolean
37 {
38     return self.type.name.toUpperCase().contains("ATTRIBUTES");
39 }
```

Properties Problems Console

Epsilon  
Header : Data  
Entity : Food  
Attributes : FoodID  
Attributes : FoodName  
Attributes : FoodPicture  
Attributes : FoodOverAllRating  
Attributes : Description  
Attributes : Price  
Attributes : Currency  
Attributes : Category  
Attributes : Ingredients  
Entity : Review  
Attributes : ReviewID  
Attributes : ReviewText  
Attributes : ReviewRating  
Attributes : ReviewPicture  
Attributes : FoodID

Also, find attached the videos of the outputs of our models.

[EVL:EOL short video clips](#)