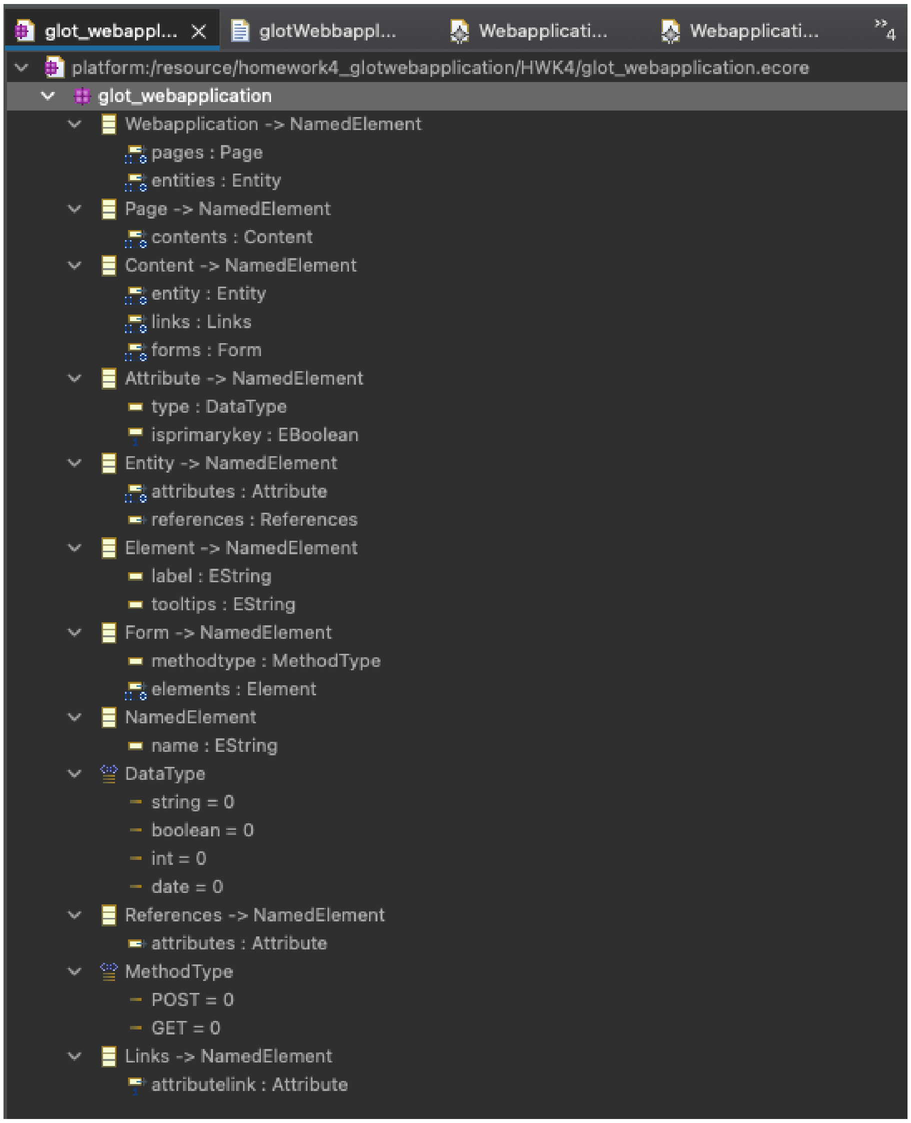
Assignment A4

Team - Transformers: Nsidibe Bassey, Niurguiana Borisova, Robel Amare Gebrewold.

Task A4.1

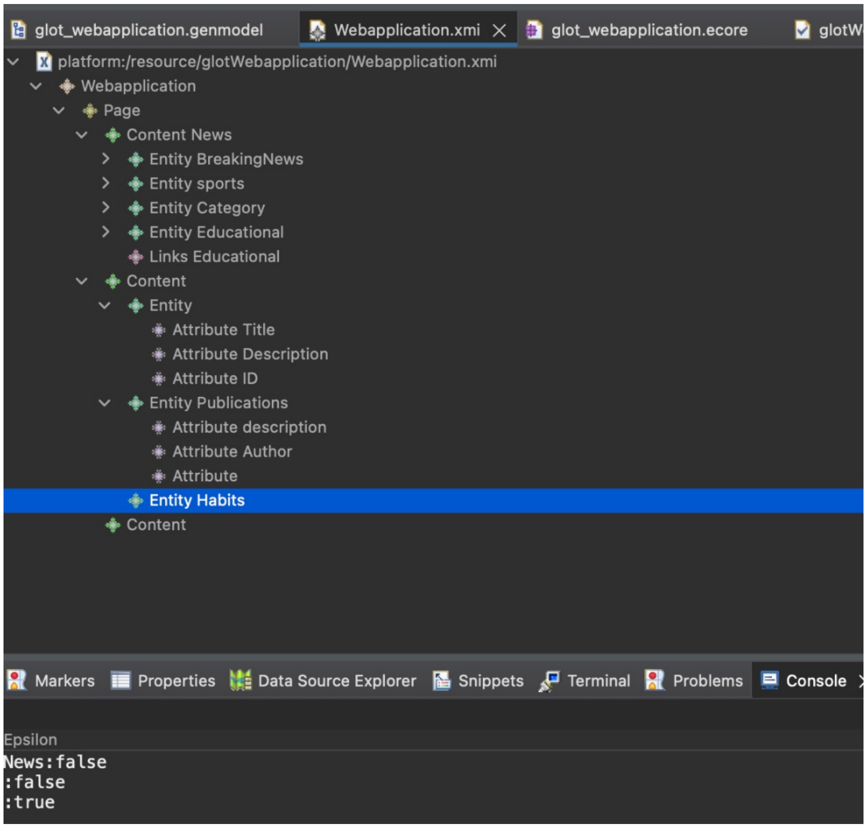
We defined a metamodel with 12 metaclasses from previous MPS homework.



```
platform:/resource/homework4_glotwebapplication/HWK4/glot_webapplication.ecore
  glot_webapplication
    Webapplication -> NamedElement
      pages : Page
      entities : Entity
    Page -> NamedElement
      contents : Content
    Content -> NamedElement
      entity : Entity
      links : Links
      forms : Form
    Attribute -> NamedElement
      type : DataType
      isprimarykey : EBoolean
    Entity -> NamedElement
      attributes : Attribute
      references : References
    Element -> NamedElement
      label : EString
      tooltips : EString
    Form -> NamedElement
      methodtype : MethodType
      elements : Element
    NamedElement
      name : EString
    DataType
      string = 0
      boolean = 0
      int = 0
      date = 0
    References -> NamedElement
      attributes : Attribute
    MethodType
      POST = 0
      GET = 0
    Links -> NamedElement
      attributelink : Attribute
```

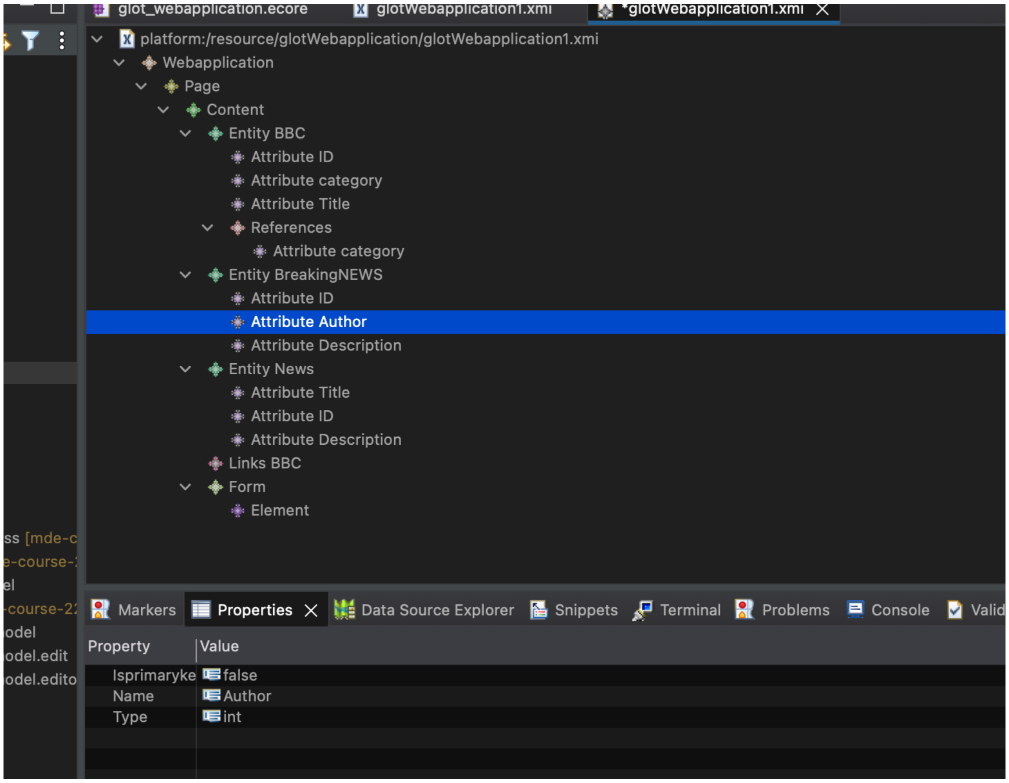
Task A4.2

Instances of a model: Webapplication.xmi and glotwebapplication1.xmi.



```
platform:/resource/glotWebapplication/Webapplication.xmi
  Webapplication
    Page
      Content News
        Entity BreakingNews
        Entity sports
        Entity Category
        Entity Educational
        Links Educational
      Content
        Entity
          Attribute Title
          Attribute Description
          Attribute ID
        Entity Publications
          Attribute description
          Attribute Author
          Attribute
        Entity Habits
      Content
```

```
Epsilon
News:false
:false
:true
```

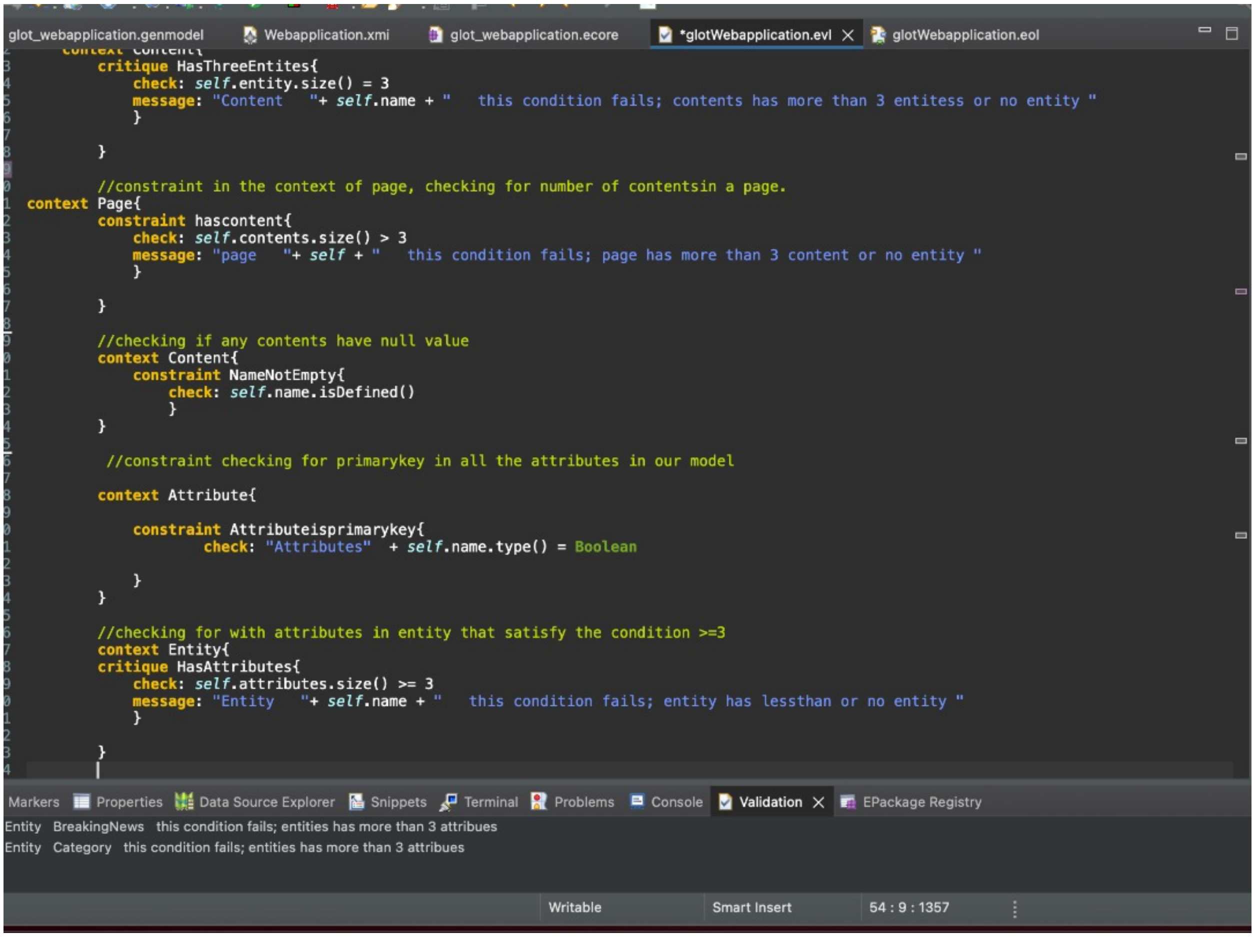


```
platform:/resource/glotWebapplication/glotWebapplication1.xmi
  Webapplication
    Page
      Content
        Entity BBC
          Attribute ID
          Attribute category
          Attribute Title
          References
            Attribute category
        Entity BreakingNEWS
          Attribute ID
          Attribute Author
          Attribute Description
        Entity News
          Attribute Title
          Attribute ID
          Attribute Description
        Links BBC
        Form
          Element
```

Property	Value
Isprimaryke	false
Name	Author
Type	int

Task A4.3

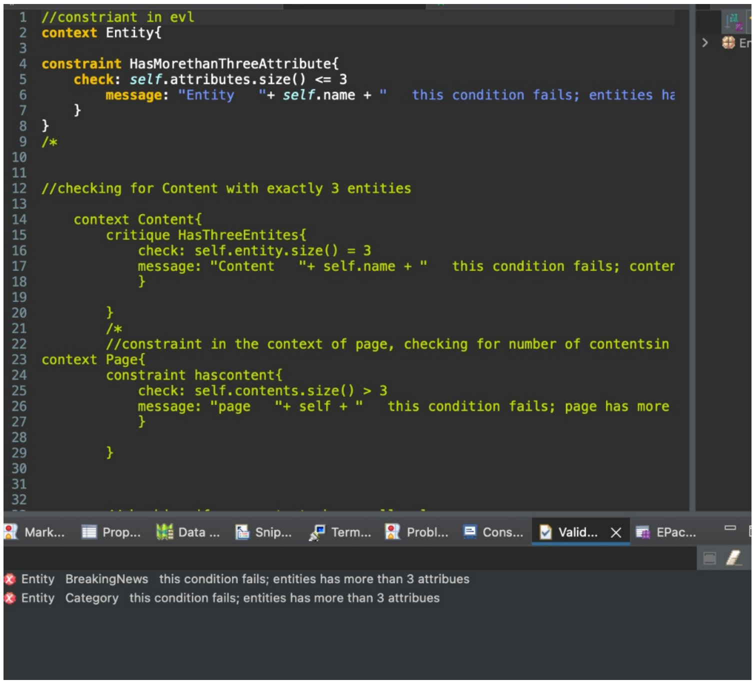
3 metamodel constraints (EVL):



```
    context Content{
        critique HasThreeEntites{
            check: self.entity.size() = 3
            message: "Content   "+ self.name + "   this condition fails; contents has more than 3 entitess or no entity "
            }

        }

        //constraint in the context of page, checking for number of contentsin a page.
context Page{
        constraint hascontent{
            check: self.contents.size() > 3
            message: "page   "+ self + "   this condition fails; page has more than 3 content or no entity "
            }

        }

        //checking if any contents have null value
        context Content{
            constraint NameNotEmpty{
                check: self.name.isDefined()
                }
        }

         //constraint checking for primarykey in all the attributes in our model

        context Attribute{

            constraint Attributeisprimarykey{
                    check: "Attributes"  + self.name.type() = Boolean

            }
        }

        //checking for with attributes in entity that satisfy the condition >=3
        context Entity{
        critique HasAttributes{
            check: self.attributes.size() >= 3
            message: "Entity   "+ self.name + "   this condition fails; entity has lessthan or no entity "
            }

        }
```

```
Entity   BreakingNews   this condition fails; entities has more than 3 attribues
Entity   Category   this condition fails; entities has more than 3 attribues
```

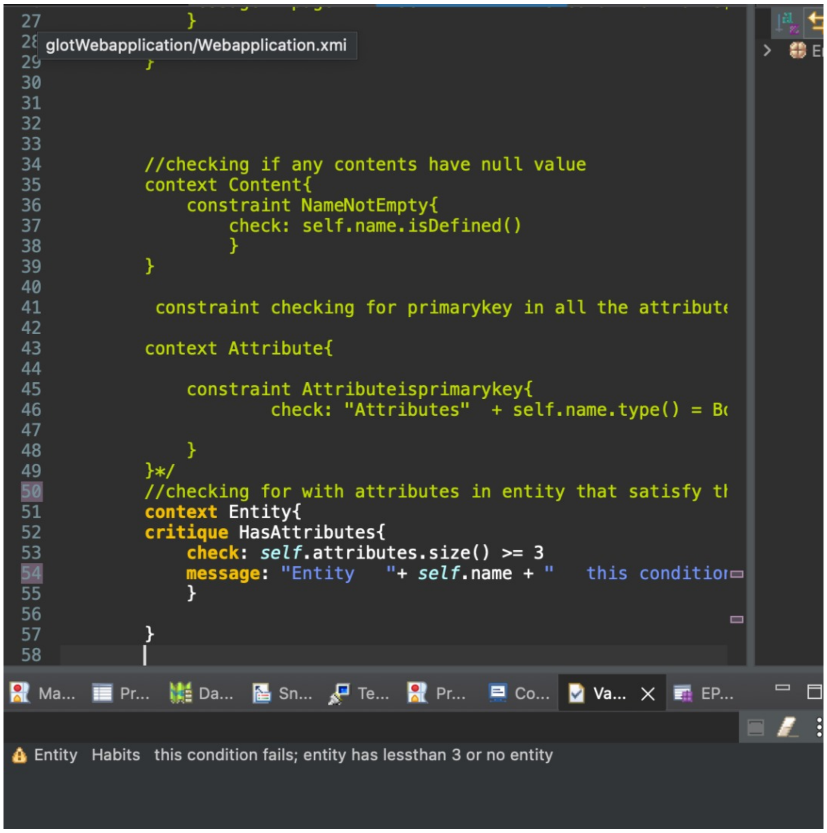
1) Checking if Entity has more than 3 Attributes:



```
//constriant in evl
context Entity{

constraint HasMorethanThreeAttribute{
    check: self.attributes.size() <= 3
        message: "Entity   "+ self.name + "   this condition fails; entities ha
    }
}
/*


//checking for Content with exactly 3 entities

    context Content{
        critique HasThreeEntites{
            check: self.entity.size() = 3
            message: "Content   "+ self.name + "   this condition fails; conter
            }

        }
        /*
        //constraint in the context of page, checking for number of contentsin
context Page{
        constraint hascontent{
            check: self.contents.size() > 3
            message: "page   "+ self + "   this condition fails; page has more
            }

        }
```

```
Entity   BreakingNews   this condition fails; entities has more than 3 attribues
Entity   Category   this condition fails; entities has more than 3 attribues
```

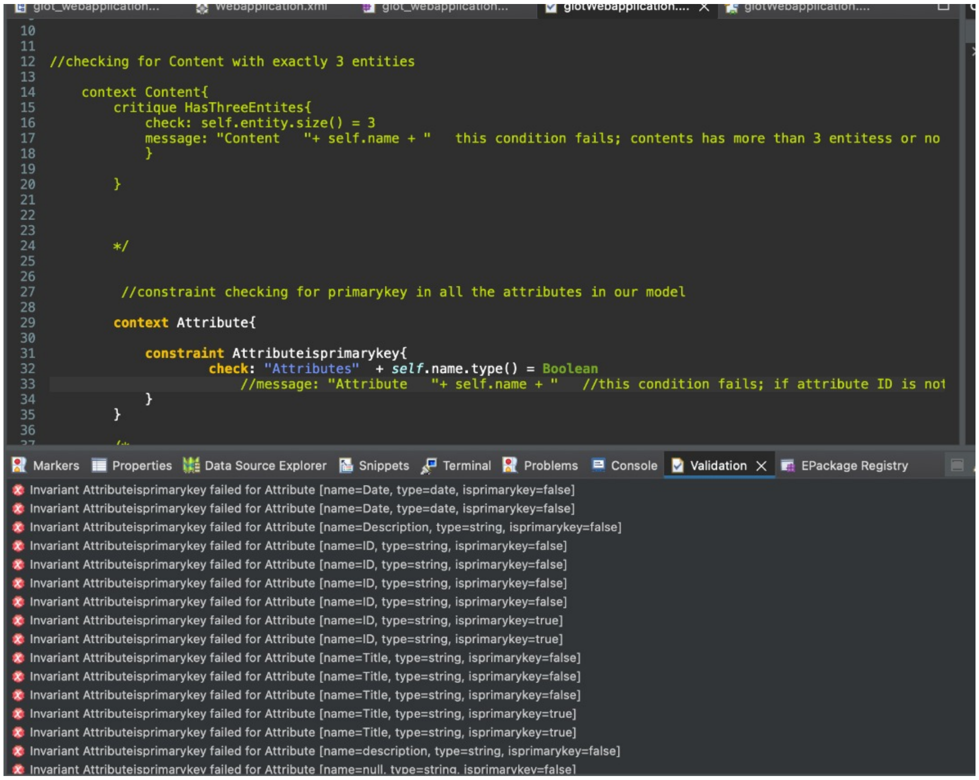
2) Checking if content has 3 entities:



```
                }

            }




        //checking if any contents have null value
        context Content{
            constraint NameNotEmpty{
                check: self.name.isDefined()
                }
        }

         constraint checking for primarykey in all the attribute

        context Attribute{

            constraint Attributeisprimarykey{
                    check: "Attributes"  + self.name.type() = Bo

            }
        }*/
        //checking for with attributes in entity that satisfy tl
        context Entity{
        critique HasAttributes{
            check: self.attributes.size() >= 3
            message: "Entity   "+ self.name + "   this conditio
            }

        }
        |
```

```
Entity   Habits   this condition fails; entity has lessthan 3 or no entity
```

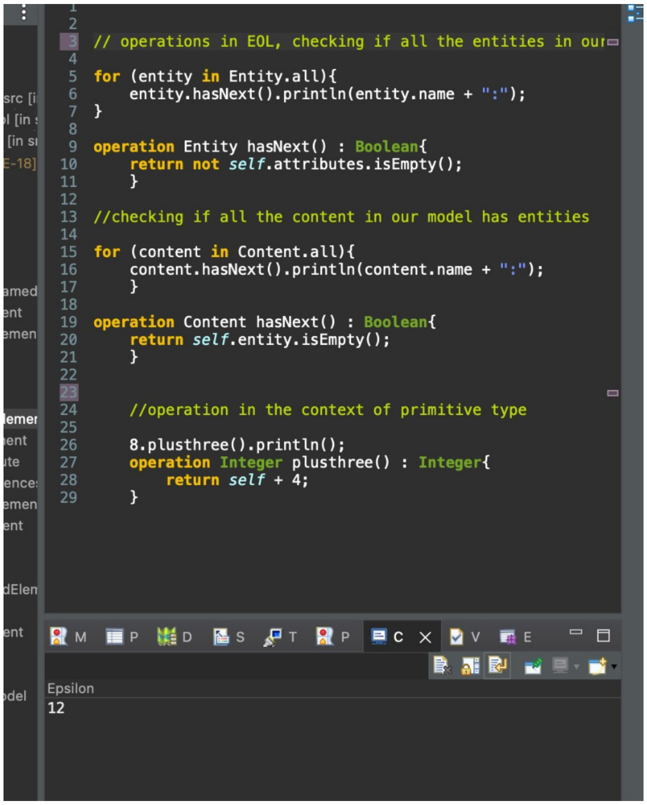
3) Checking for Primary Key in all the attributes in our model:



```


//checking for Content with exactly 3 entities

    context Content{
        critique HasThreeEntites{
            check: self.entity.size() = 3
            message: "Content   "+ self.name + "   this condition fails; contents has more than 3 entitess or no
            }

        }



        */


         //constraint checking for primarykey in all the attributes in our model

        context Attribute{

            constraint Attributeisprimarykey{
                    check: "Attributes"  + self.name.type() = Boolean
                        //message: "Attribute   "+ self.name + "   //this condition fails; if attribute ID is not
            }
        }

```

```
Invariant Attributeisprimarykey failed for Attribute [name=Date, type=date, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=Date, type=date, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=Description, type=string, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=ID, type=string, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=ID, type=string, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=ID, type=string, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=ID, type=string, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=ID, type=string, isprimarykey=true]
Invariant Attributeisprimarykey failed for Attribute [name=ID, type=string, isprimarykey=true]
Invariant Attributeisprimarykey failed for Attribute [name=Title, type=string, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=Title, type=string, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=Title, type=string, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=Title, type=string, isprimarykey=true]
Invariant Attributeisprimarykey failed for Attribute [name=Title, type=string, isprimarykey=true]
Invariant Attributeisprimarykey failed for Attribute [name=description, type=string, isprimarykey=false]
Invariant Attributeisprimarykey failed for Attribute [name=null, type=string, isprimarykey=false]
```

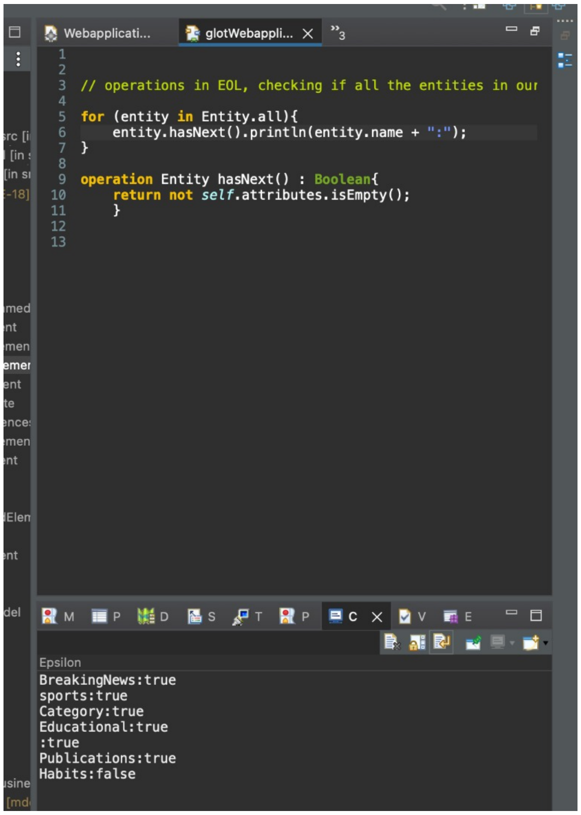
3 operations (EOL):



```

// operations in EOL, checking if all the entities in our

for (entity in Entity.all){
    entity.hasNext().println(entity.name + ":");
}

operation Entity hasNext() : Boolean{
    return not self.attributes.isEmpty();
    }

//checking if all the content in our model has entities

for (content in Content.all){
    content.hasNext().println(content.name + ":");
    }

operation Content hasNext() : Boolean{
    return self.entity.isEmpty();
    }


    //operation in the context of primitive type

    8.plusthree().println();
    operation Integer plusthree() : Integer{
        return self + 4;
    }
```

```
Epsilon
12
```

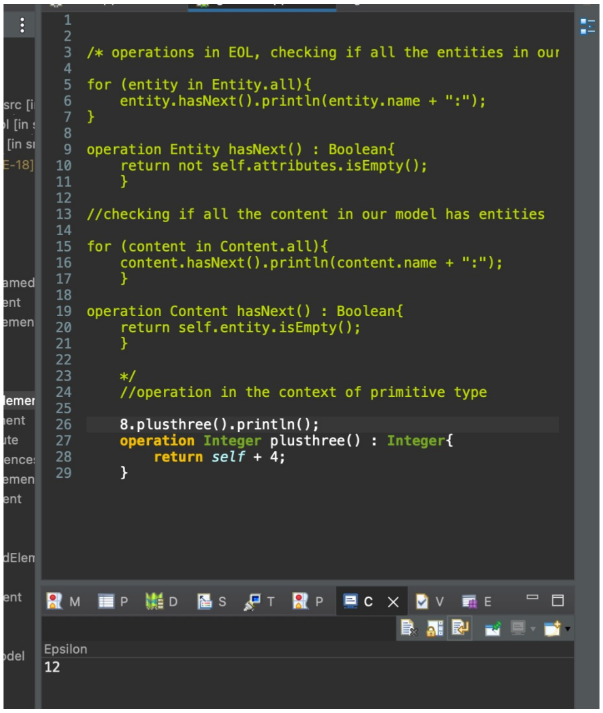
1) First one is checking Attributes inside Entities if they are empty:



```
// operations in EOL, checking if all the entities in our

for (entity in Entity.all){
    entity.hasNext().println(entity.name + ":");
}

operation Entity hasNext() : Boolean{
    return not self.attributes.isEmpty();
    }
```

```
Epsilon
BreakingNews:true
sports:true
Category:true
Educational:true
:true
Publications:true
Habits:false
```

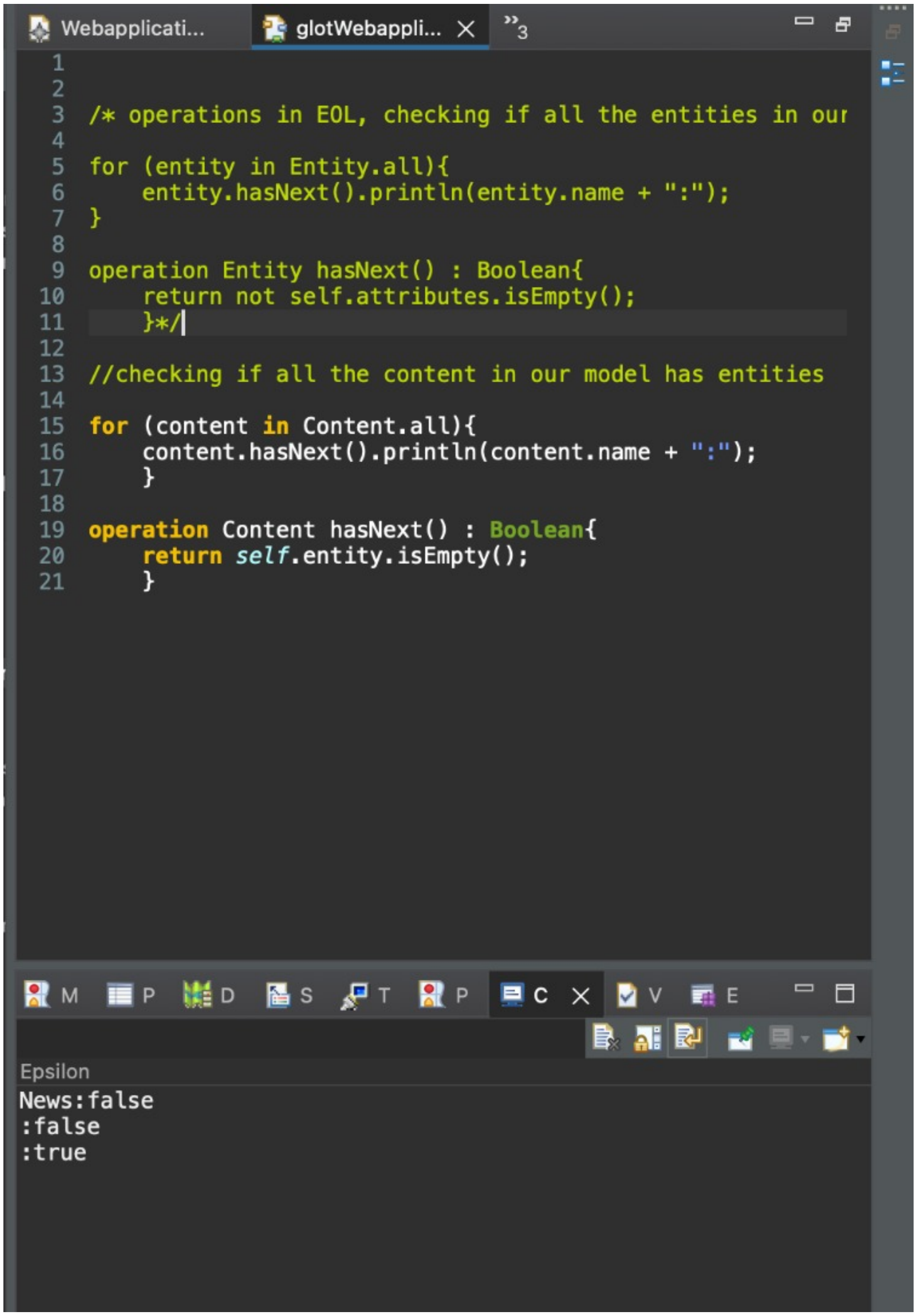
2) Second one is operation in the context of primitive type:



```
/* operations in EOL, checking if all the entities in our

for (entity in Entity.all){
    entity.hasNext().println(entity.name + ":");
}

operation Entity hasNext() : Boolean{
    return not self.attributes.isEmpty();
    }

//checking if all the content in our model has entities

for (content in Content.all){
    content.hasNext().println(content.name + ":");
    }

operation Content hasNext() : Boolean{
    return self.entity.isEmpty();
    }

    */
    //operation in the context of primitive type

    8.plusthree().println();
    operation Integer plusthree() : Integer{
        return self + 4;
    }
```

```
Epsilon
12
```

3) Third one is checking if all the content in our model has entities:



```


/* operations in EOL, checking if all the entities in our

for (entity in Entity.all){
    entity.hasNext().println(entity.name + ":");
}

operation Entity hasNext() : Boolean{
    return not self.attributes.isEmpty();
    }*/

//checking if all the content in our model has entities

for (content in Content.all){
    content.hasNext().println(content.name + ":");
    }

operation Content hasNext() : Boolean{
    return self.entity.isEmpty();
    }
```

```
News:false
:false
:true
```