MODEL DRIVEN ENGINEERING - ASSIGNMENT 4

TEAM: Supercalifragilisticexpialidocious
APOORVA NALINI PRADEEP KUMAR
JONATHAN THANGADURAI SELVARAJ

TASK A4.1 - Definition of Metamodel

The following metaclasses are defined in the metamodel to cover the different aspects of “Glot” a domain specific language for specifying Web Applications.

- **NamedElement** → Metaclass for specifying names.
- **WebApplication** → Root metaclass in which all parts of the web application are specified.
- **Entity** → Metaclass for specifying **Attribute** and **Reference** contained in **Page** and **Form.**
- **Feature** → Abstract metaclass which is used to describe different features of the **Entity.**
- **Attribute** → Metaclass which describes the characteristics or properties of an **Entity.**
- **Reference** → Metaclass that extends **Feature** and houses foreignkey reference to the other **Entity.**
- **Page** → Metaclass which describes a page in a **WebApplication** and has **Content** in them.
- **Content** → Abstract metaclass which describes the contents which are displayed in a **Page.**
- **StaticContent** → Abstract metaclass for representing the static contents in a **Page.**
- **Form** → Metaclass illustrating a web form that gets values from the user.
- **Element** → Metaclass which represents the web elements such as checkboxes, buttons, etc in **Form.**
- **Media** → Metaclass dedicated for media elements like image, video, etc.
- **DynamicContent** → Abstract metaclass for representing the dynamic contents in a **Page.**
- **Index** → Metaclass for describing dynamic contents as a list of items in a **Page.**
- **Individual** → Metaclass for specifying the contents of individual items in a details **Page.**

The enumerations used by the above metaclasses are:

- **DataType** → Specifies the data types of attributes.
- **PartOfPage** → Specifies the position of content in a Page.
- **MethodType** → Specifies the HTTP Method for getting data from the Form.
- **ElementType** → Defines the types of elements in the Form.
- **MediaType** → Defines the type of media content.

Custom Data Type → A custom type **java.time.Instant** has been introduced to get createdAt value for a Page.

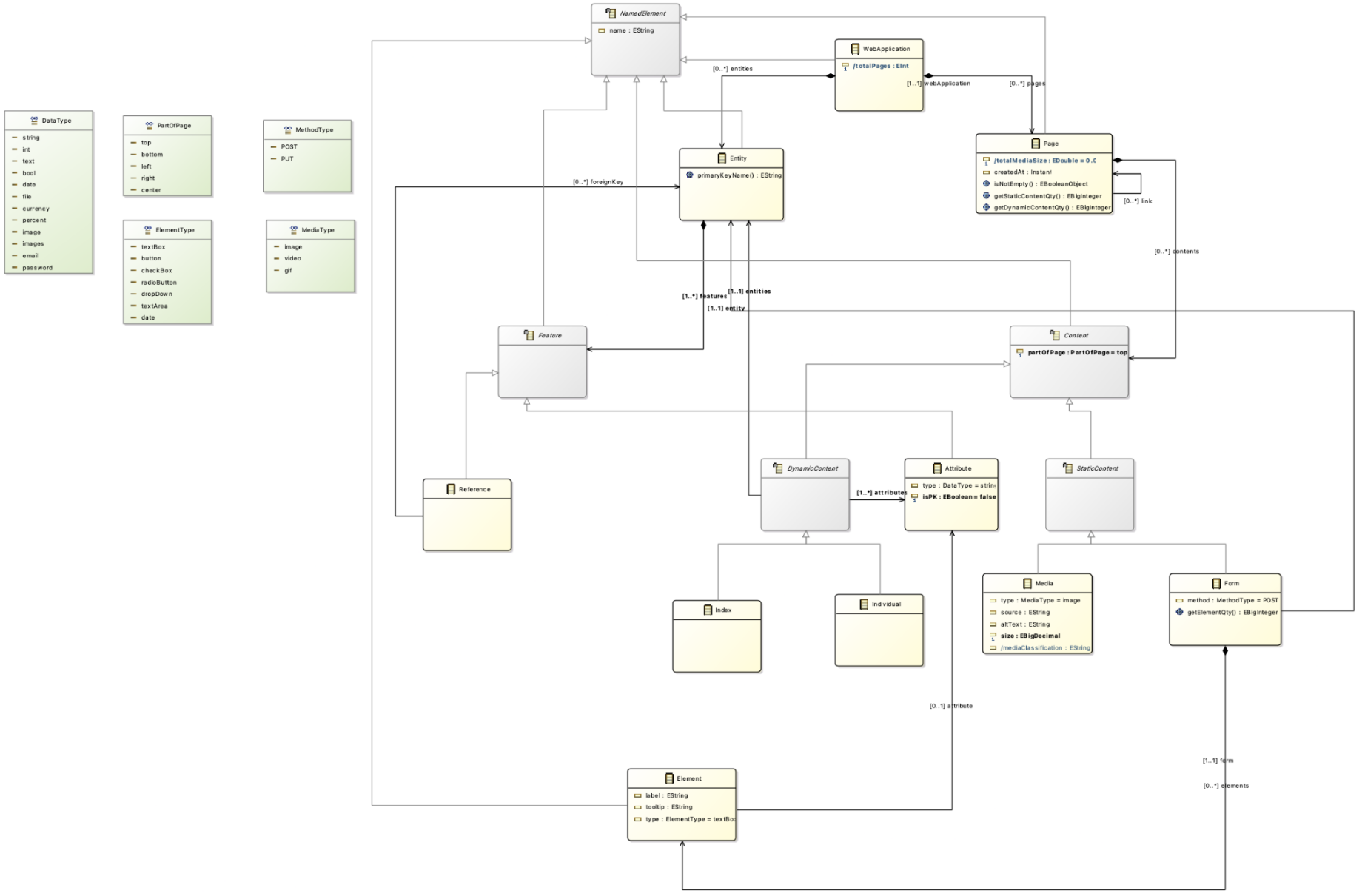
All the metaclasses classes **inherit** from the metaclass “NamedElement”. “WebApplication” **contains** “Page” and “Entity”. “Entity” **contains** a list of “Feature” (abstract class) and it is the **base class** for “Attribute” and “Reference”. “Reference” **references** “Entity”.

“Page” can have links (**reference)** to other “Page”. “Page” **contains** “Content” which is an **abstract class.** “Static Content” and “Dynamic Content” both **inherit** from “Content”. An **opposite** relationship (bi-directional reference) exists between “Web Application” and “Page” - where a “Web Application” **contains zero or many** “Page” and each “Page” is **part of one** “Web Application”.

“Static Content” is an **abstract class** and “Form” and “Media” **inherit** from “Static Content”. “Form” **references** “Entity” and **contains zero or many** “Element”. “Element” **references** “Attribute”. An **opposite** relationship (bi-directional reference) also exists between “Form” and “Element” - where a “Form” **contains zero or many** “Element” and each “Element” is **part of one** “Form”.

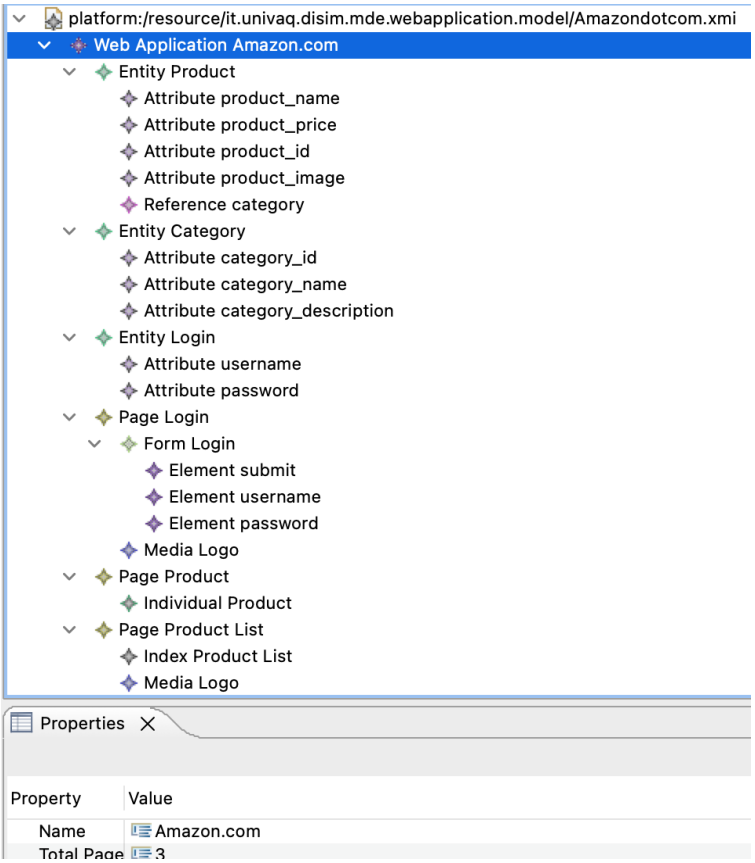
“Dynamic Content” is an **abstract class** that **references** “Entity” and “Attribute”. “Index” and “Individual” **inherit** from “Dynamic Content”.

The following Ecore diagram illustrates the above relationships:


NamedElement
name : EString
WebApplication
/totalPages : EInt
[0..*] entities
[1..1] webApplication
[0..*] pages
Page
/totalMediaSize : EDouble = 0.0
createdAt : Instant
isNotEmpty() : EBooleanObject
getStaticContentQty() : EBigInteger
getDynamicContentQty() : EBigInteger
[0..*] link
Entity
primaryKeyName() : EString
[0..*] foreignKey
[0..*] contents
[1..*] features
[1..1] entities
[1..1] entity
Feature
Content
partOfPage : PartOfPage = top
Reference
DynamicContent
Attribute
type : DataType = strin
isPK : EBoolean = false
[1..*] attributes
StaticContent
Index
Individual
Media
type : MediaType = image
source : EString
altText : EString
size : EBigDecimal
/mediaClassification : EString
Form
method : MethodType = POST
getElementQty() : EBigInteger
[0..1] attribute
[1..1] form
[0..*] elements
Element
label : EString
tooltip : EString
type : ElementType = textBo
DataType
string
int
text
bool
date
file
currency
percent
image
images
email
password
PartOfPage
top
bottom
left
right
center
MethodType
POST
PUT
ElementType
textBox
button
checkBox
radioButton
dropDown
textArea
date
MediaType
image
video
gif

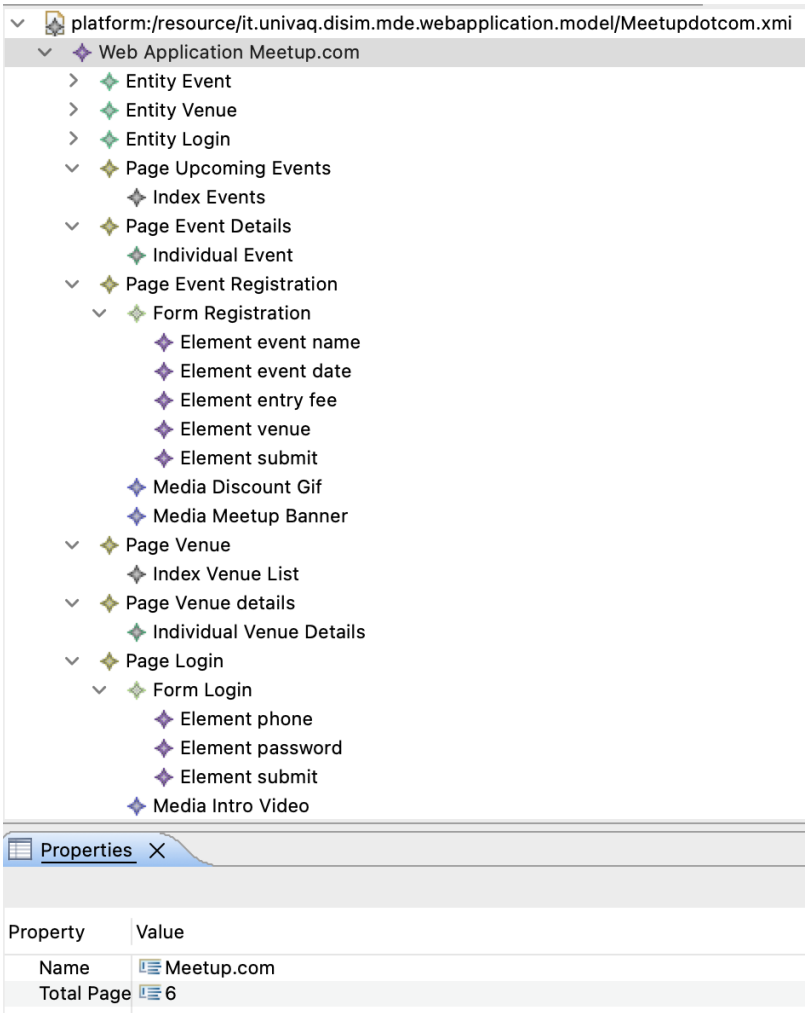
TASK A4.2 - Instantiating the metamodel with concrete instances

Concrete instance for WebApplication - Amazon.com



Amazon.com has a login page which has a form with elements - username, password and submit (Entity Login). The Page also has a media image - Logo.
The Page Product List contains information on a list of products (product_name, product_price, product_image) which are displayed on the Page Product List. This page also has a media image - logo. The Page Product contains individual content for a Product and has attributes - product_name, product_price, category_name, category_description, product_image.

Concrete instance for WebApplication - Meetup.com

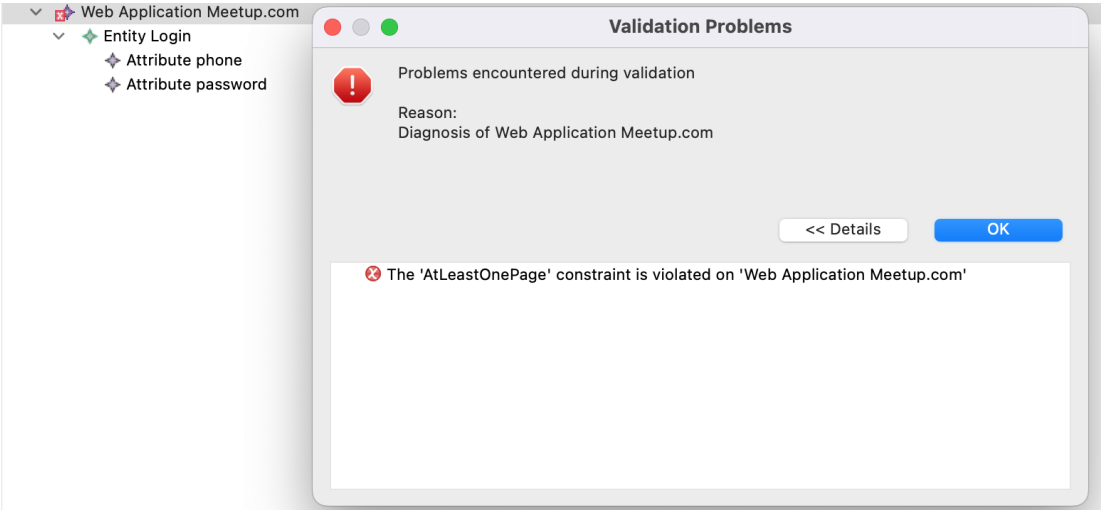


Meetup.com has a Login page which has a form with elements - phone, password (Entity Login) and submit. The Page also has a media - Intro video.
The Page Event Registration has a form with the elements - event name, event date, entry fee, venue, submit and also has media - discount gif and meetup banner. The Upcoming Events Page has index Content - Events for displaying a list of events which comprises event_name, event_date and event_entryFee.
The Venue Page has index Content - Venue List for displaying a list of venues which comprises venue_name and venue_address.
The Venue Details Page has Content - Individual named Venue Details which displays details of the individual venues and events happening at that particular venue. It comprises venue_name, venue_address, event_name, event_date, event_entryFee.
The Event Details Page has Content - Individual named Event that displays details of individual events and comprises event_name, event_date and event_entryFee, venue_address and venue_name.

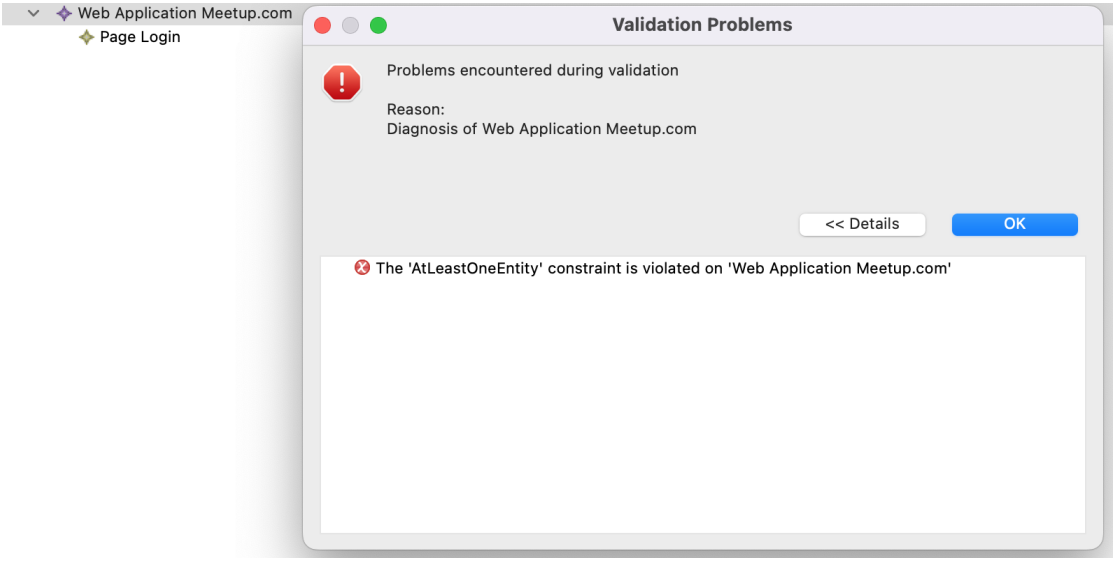
TASK A4.3 - Metamodel constraints, operations and derived fields in OCL

The following **constraints** were introduced in the WebApplication

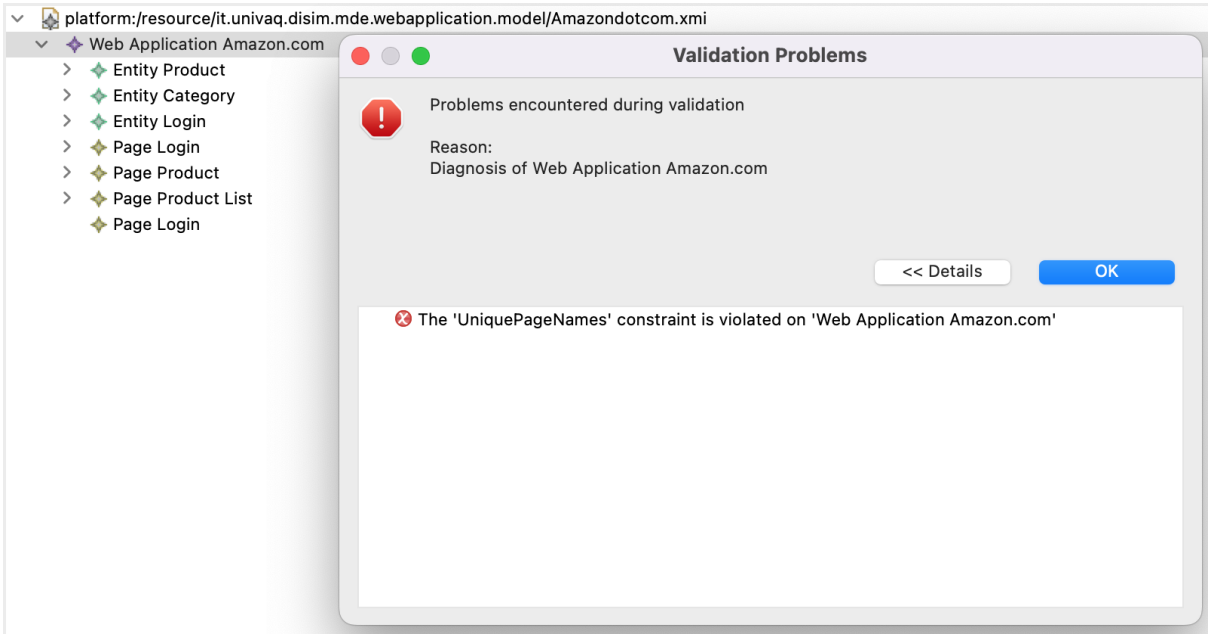
- AtLeastOnePage → For WebApplication
- AtLeastOneEntity → For WebApplication
- UniquePageNames → For WebApplication
- UniqueEntityNames → For WebApplication
- UniqueAttributeReferenceNames → For Entity
- OnePrimaryKey → For Entity
- OptimisedPageLoad → For Page
- AtLeastOneElement → For Form



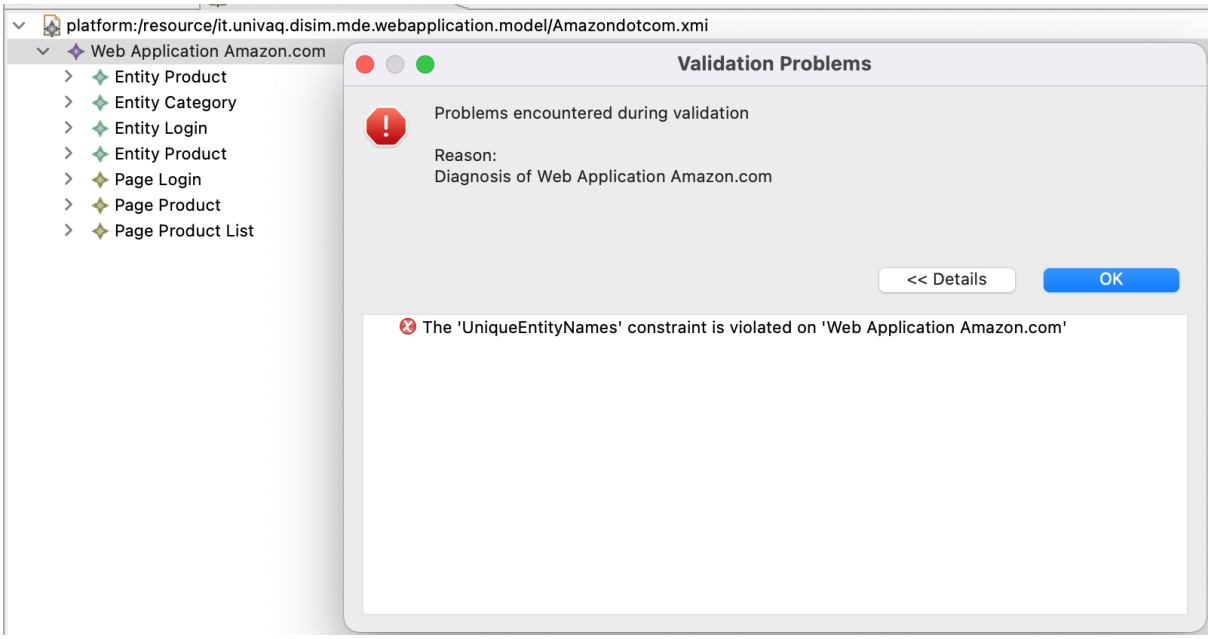
AtLeastOnePage Constraint
(The WebApplication needs to have at least one page)



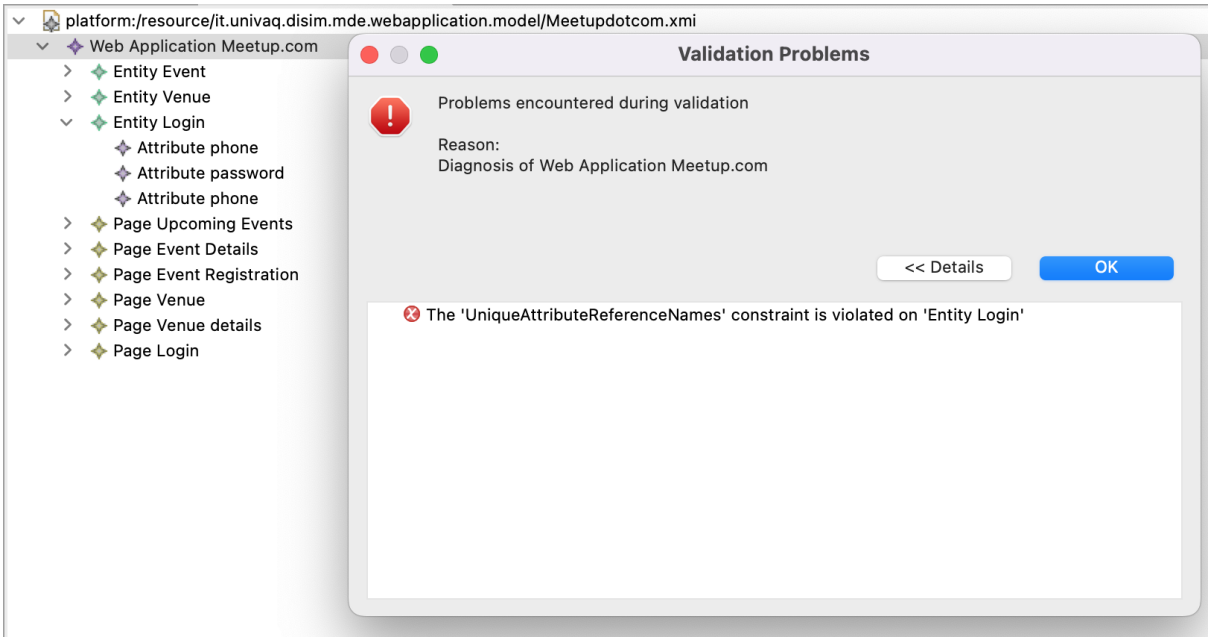
AtLeastOneEntity Constraint
(The WebApplication needs to have at least one Entity)



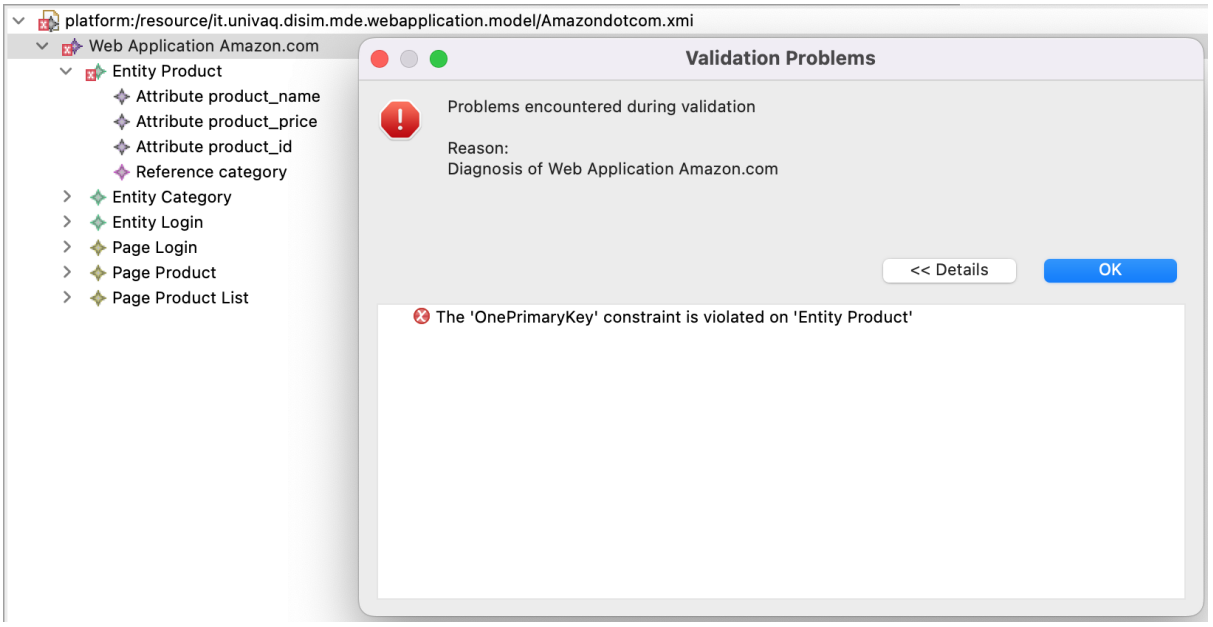
UniquePageNames Constraint
(The name of each Page in a WebApplication should be unique)



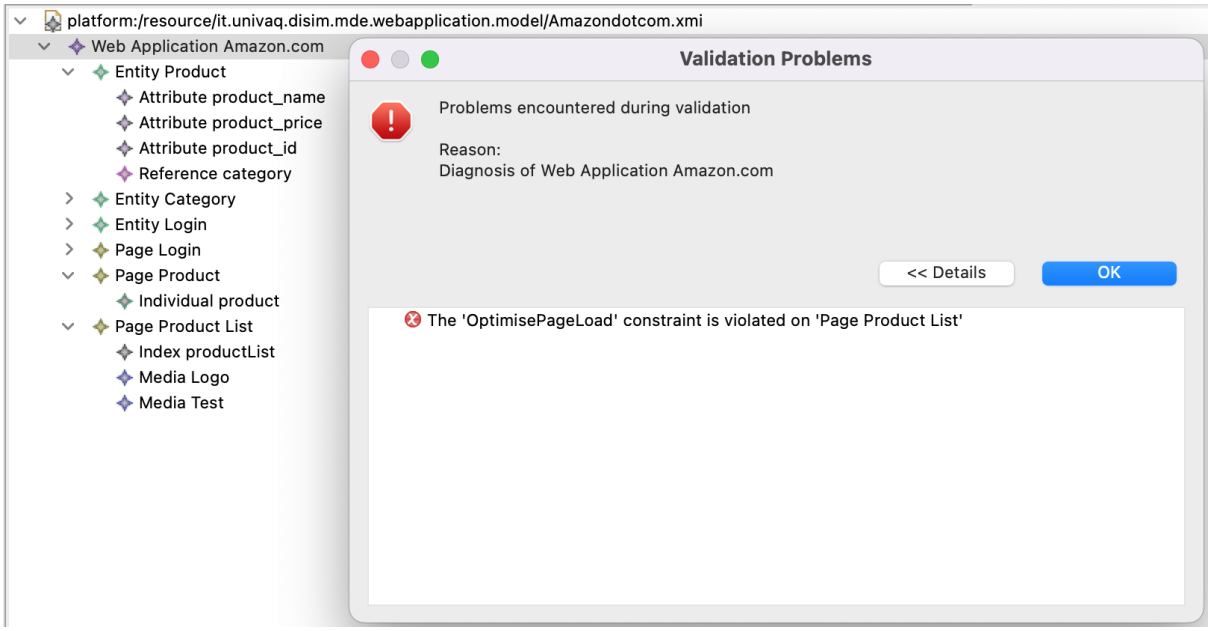
UniqueEntityNames Constraint
(The name of each Entity in a WebApplication should be unique)



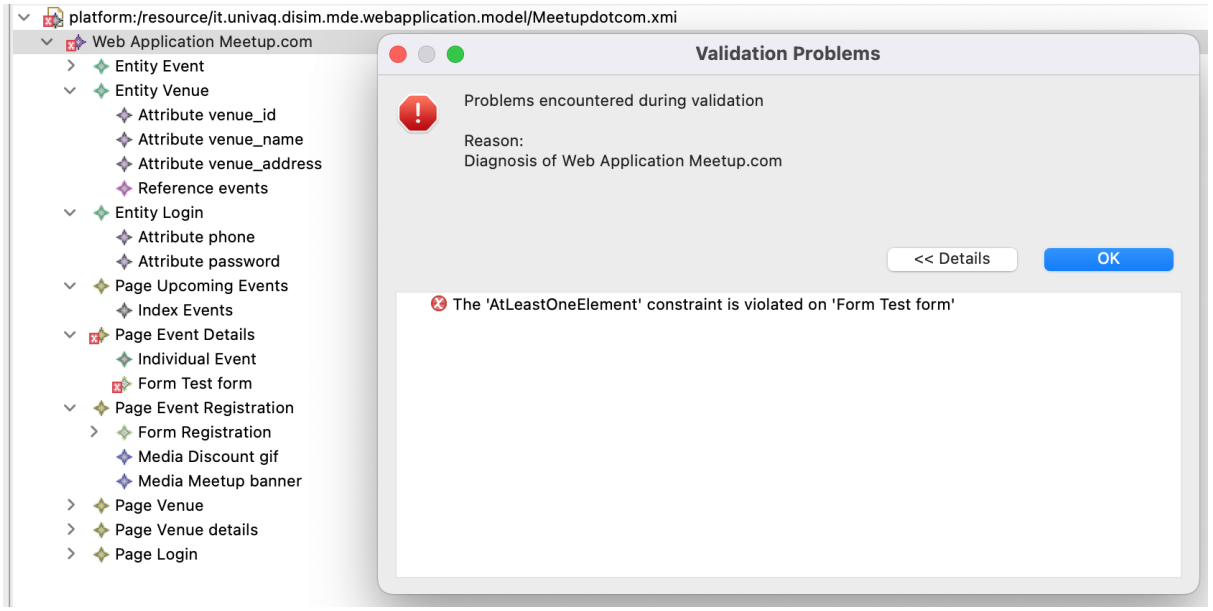
UniqueAttributeReferenceNames Constraint
(The name of each Attribute/Reference in an Entity should be unique)



OnePrimaryKey Constraint
(An Entity should have only one primary key)



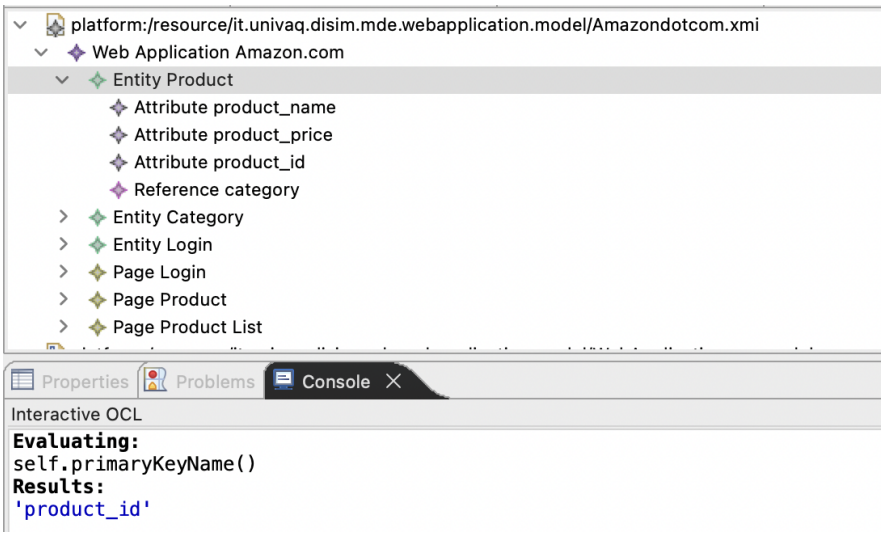
OptimisePageLoad Constraint
(The sum of sizes of all the media in a Page should be less than or equal to 50MB)



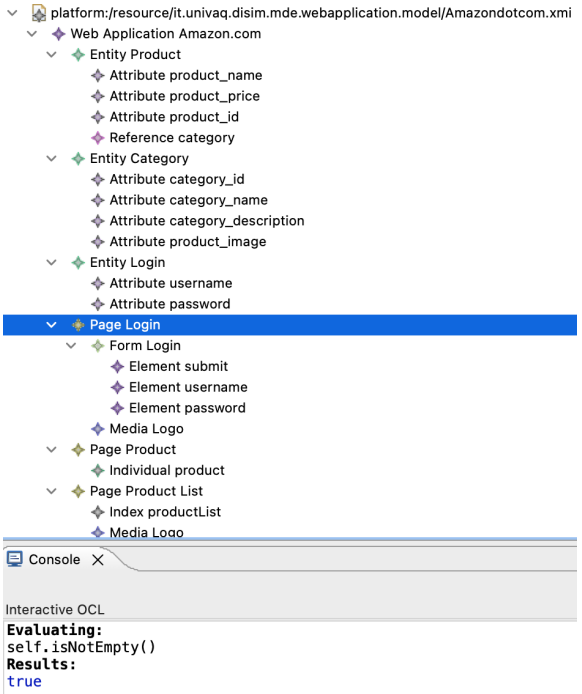
AtLeastOneElement Constraint
(The Form needs to have at least one Element)

The following **operations** were introduced in the WebApplication

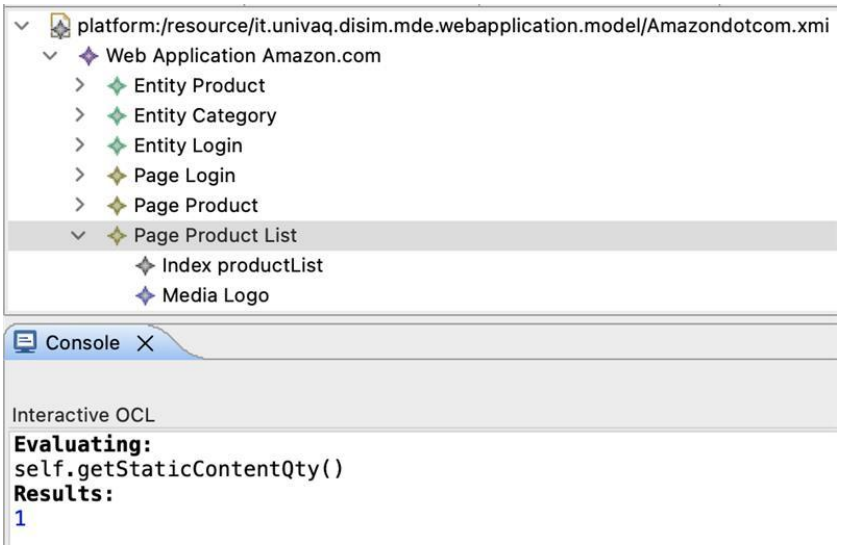
- primaryKeyName → For Entity
- isNotEmpty → For Content in Page
- getStaticContentQty → For Content in Page
- getDynamicContentQty → For Content in Page
- getElementQty → For Form



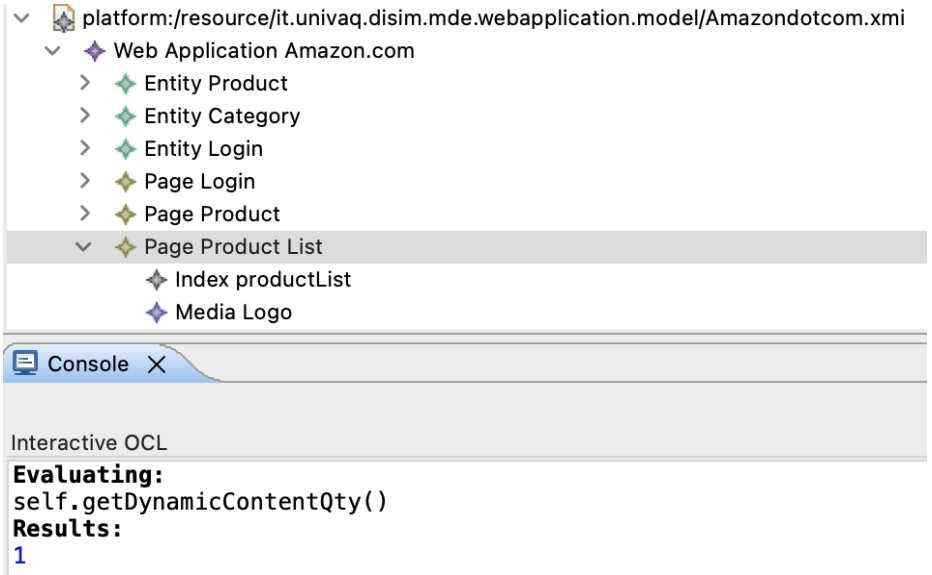
primaryKeyName Operation
(Gets the name of the primary key of the particular Entity)



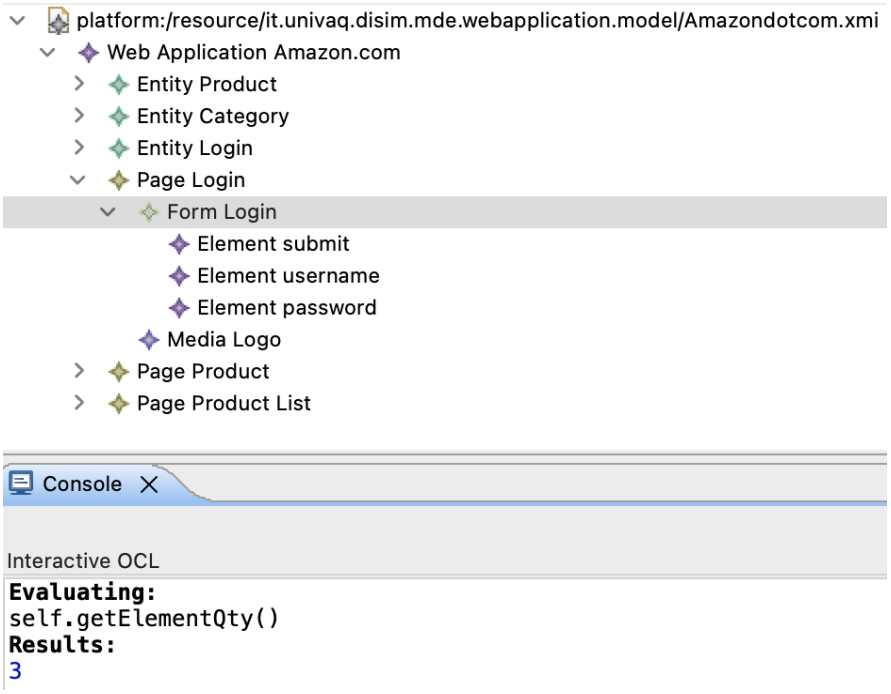
isNotEmpty Operation
(Returns true if Page has contents, else false)



getStaticContentQty Operation
(Gets the number of Static Content in a Page)



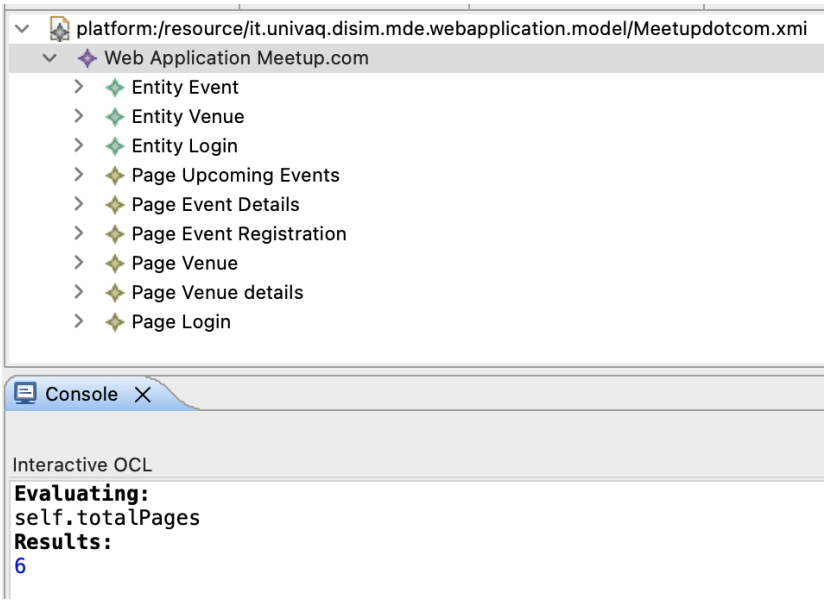
getDynamicContentQty Operation
(Gets the number of Dynamic Content in a Page)



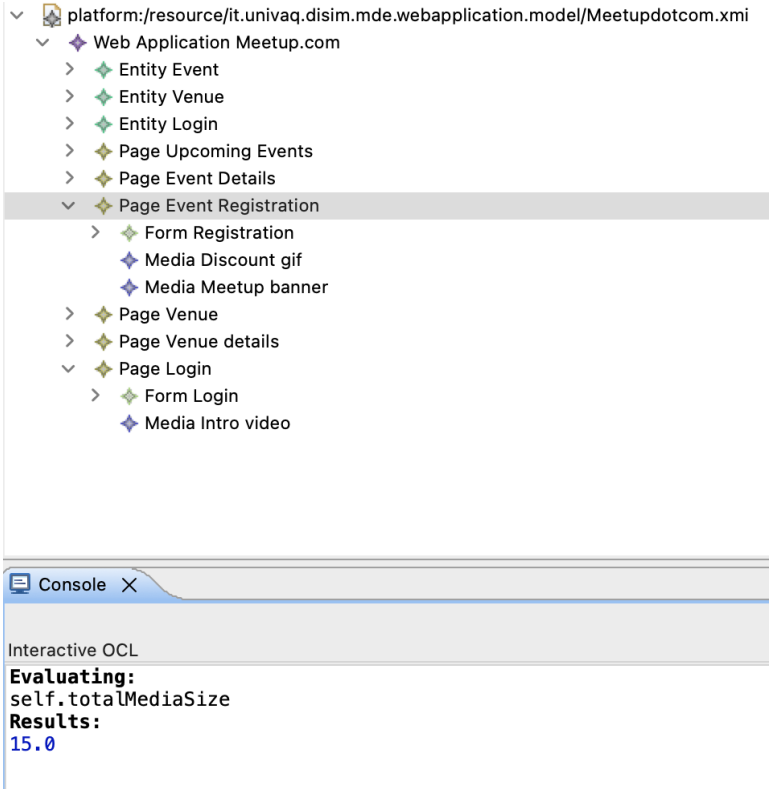
getElementQty Operation
(Gets the number of elements in a Form)

The following **derived fields** were created in the WebApplication

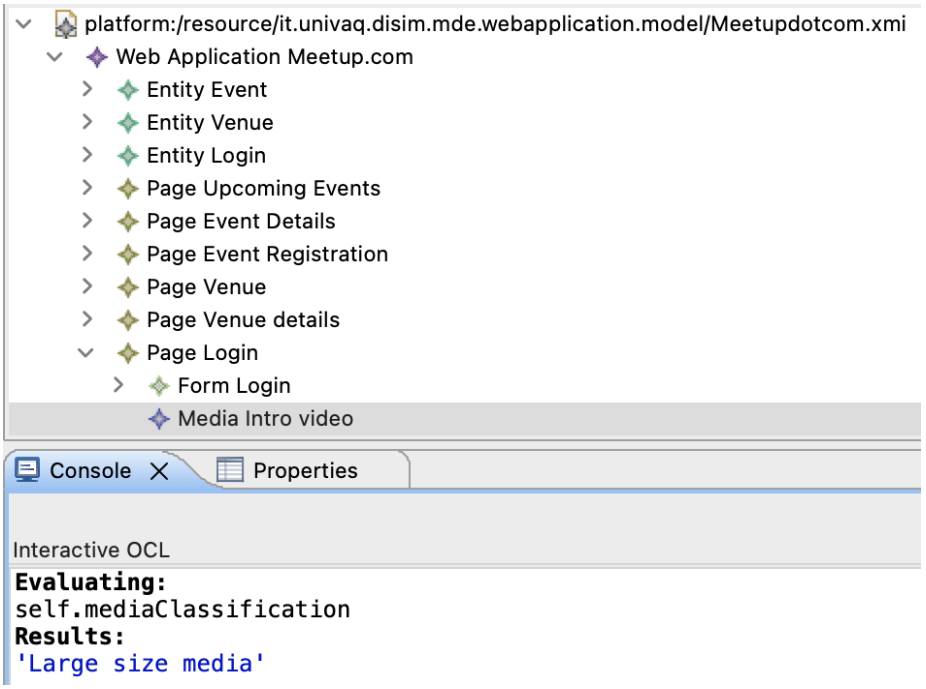
- totalPages → In WebApplication
- totalMediaSize → In Page
- mediaClassification → In Media



totalPages derived field
(Number of Pages in a WebApplication)



totalMediaSize derived field
(Has the value of sum of the size of all media in a Page in MB)



mediaClassification derived field
(Has the value 'Small size media' if the size of the media is less than 20MB, else has the value 'Large size media')