

# UNICASE

... kapsamlı bir CASE\* aracı

\* [http://en.wikipedia.org/wiki/Computer-aided\\_software\\_engineering](http://en.wikipedia.org/wiki/Computer-aided_software_engineering)

# Neden UNICASE?

- Yazılım geliştirme projelerinde yazılım mühendisliği modelleri merkezi bir yerde ve versiyon kontrolü yapılarak tutulmalıdır.
- Bahsedilen modeller gereksinimlerden, use-case ve UML modellerine, organizasyon modellerine kadar değişebilir.
- UNICASE ile tüm bu modeller birbiri ile ilişkilendirilebiliyor. Böylece yazılım geliştirme sürecini takip etmek ve yönetmek kolaylaşıyor.

# ... Neden UNICASE?

- Örneğin bir işlevsel gereksinimi gerçekleştirecek görevi tanımlayıp bu görevi ilgili gereksinime bağlayabiliyoruz. (Mesela “Kullanıcı yeni bir kural ekleyebilecek” işlev gereksinimini “Kural eklemenin gerçekleştirilmesi” görevine bağlayabiliyoruz. (Bu görevi kim yapacak vs. de yazılabiliyor.))
- Bu bağlantılar modellerin bir işbirliği içinde düzenlenmesini sağlıyor. (Mesela bir aktörün ismini değiştirdiğimizde aktörün ismi bulunduğu tüm modellerde değişiyor. Bu modellerden doküman elde ederken işlem otomatik olduğu için elle tek tek değiştirmeye gerek kalmıyor.)

# ... Neden UNICASE?

- Gereksinim analizi gibi dokümanlar müşteriye sunulacağı zaman UNICASE ile varolan modellerden model-tabanlı bir doküman oluşturmak mümkündür.
- Gereksinimlerin belirlenmesi, çözümleme yapılması, sistem ve nesne tasarımının yapılması, proje modeli üzerinde iterasyon planı, organizasyon yapısı, rational yönetim sadece bir araçla yapılabiliyor.
- Bu araç açık kaynaklı ve ücretsiz.
- ? Nightly build üstünden çalışıyor .

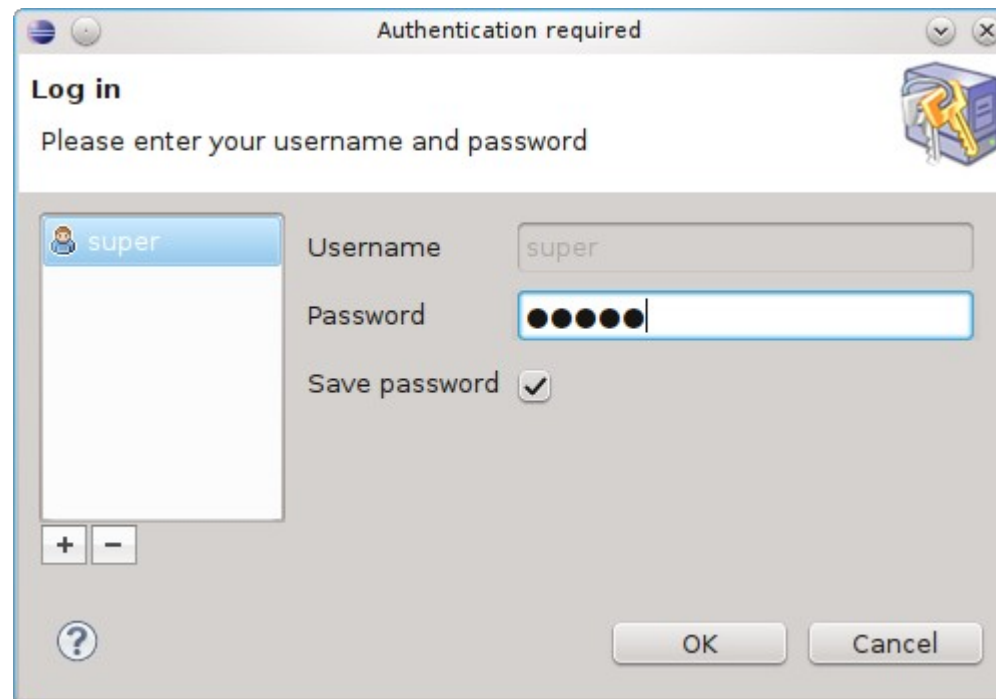
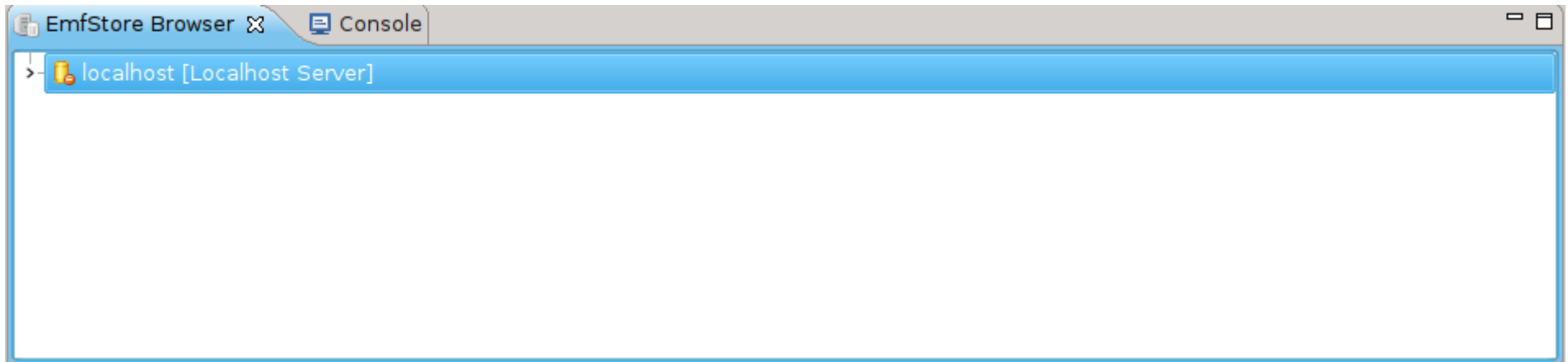
# UNICASE Nedir?

- UNICASE (Unified CASE) farklı geliştirme aktivitelerini birleştiren bir araçtır:
  - Gereksinimler
  - UML modelleri
  - Görev takvimi
  - Hata bildirimi
  - Toplantı yönetimi
  - ...

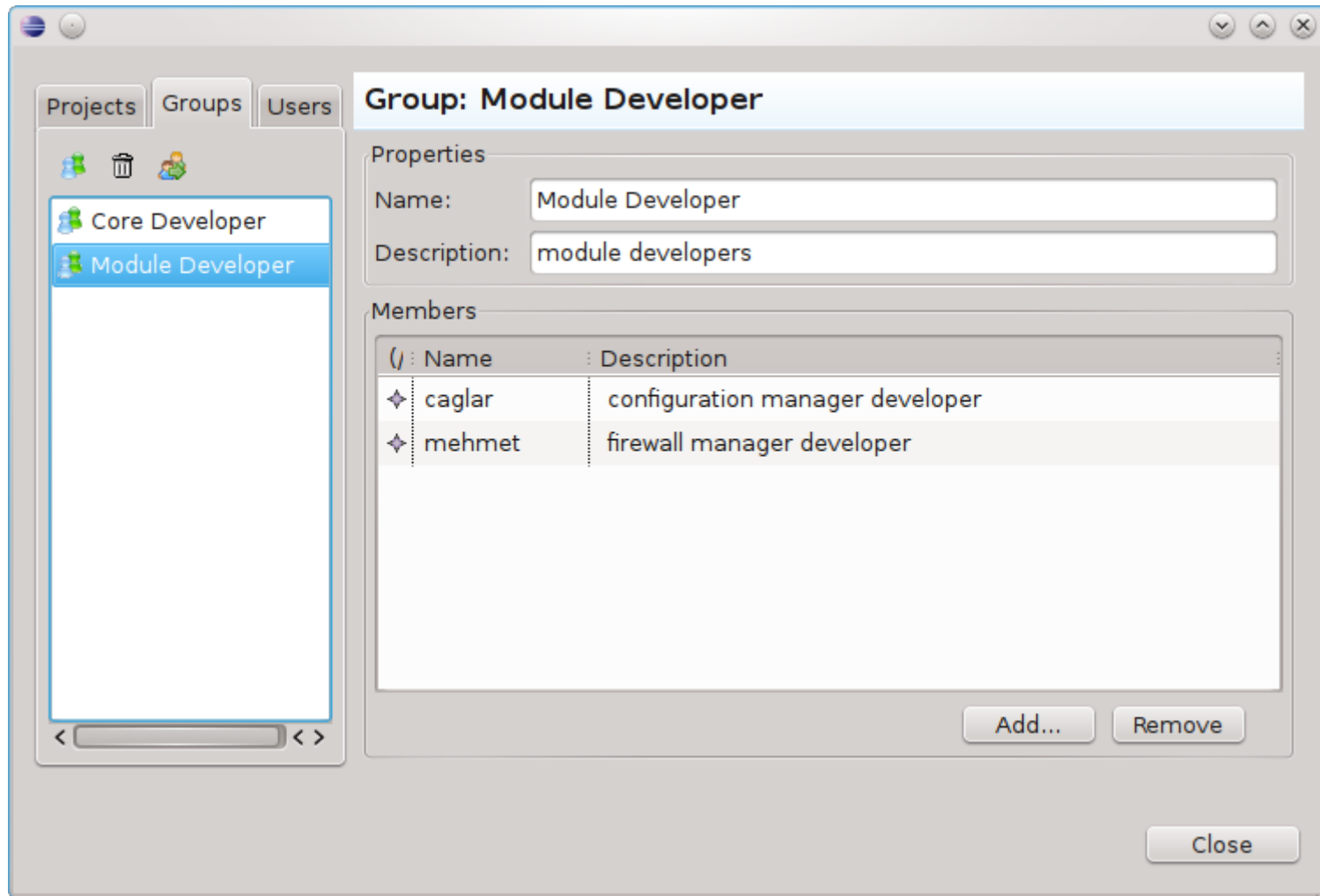
# ... UNICASE Nedir?

- UNICASE modelleri EMFStore denilen merkezi bir sunucuda tutulur.
- EMFStore SVN gibi versiyon denetimi sağlar ama modeller için özelleşmiştir.
- UNICASE Eclipse tabanlı çalışır. (EMF ve GMF gerektirir.)
- Lisansı: Eclipse Public License v 1.0 (EPL)

# EMF Store Browser & Login

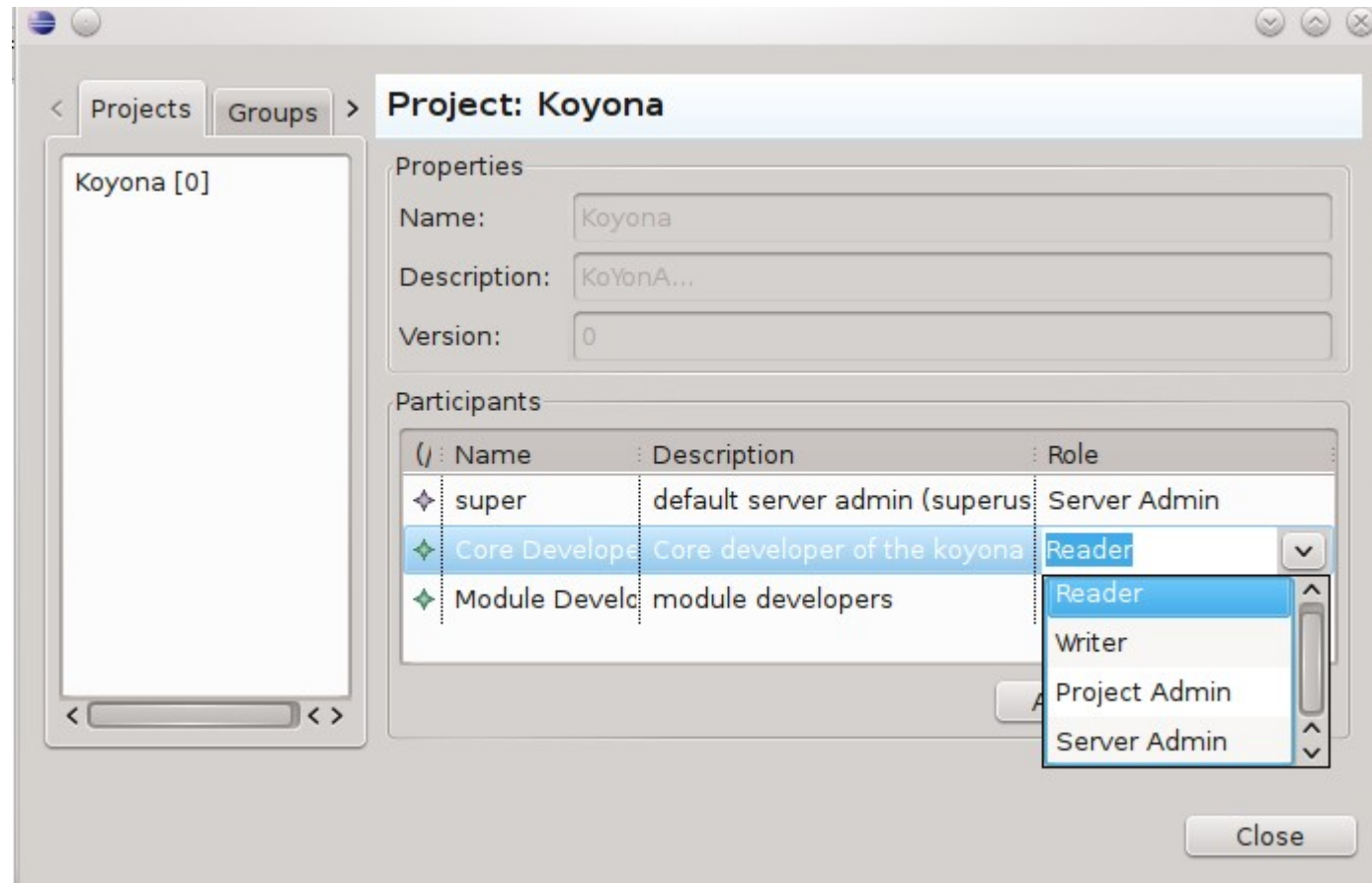


# Proje / Grup ve Kullanıcılar





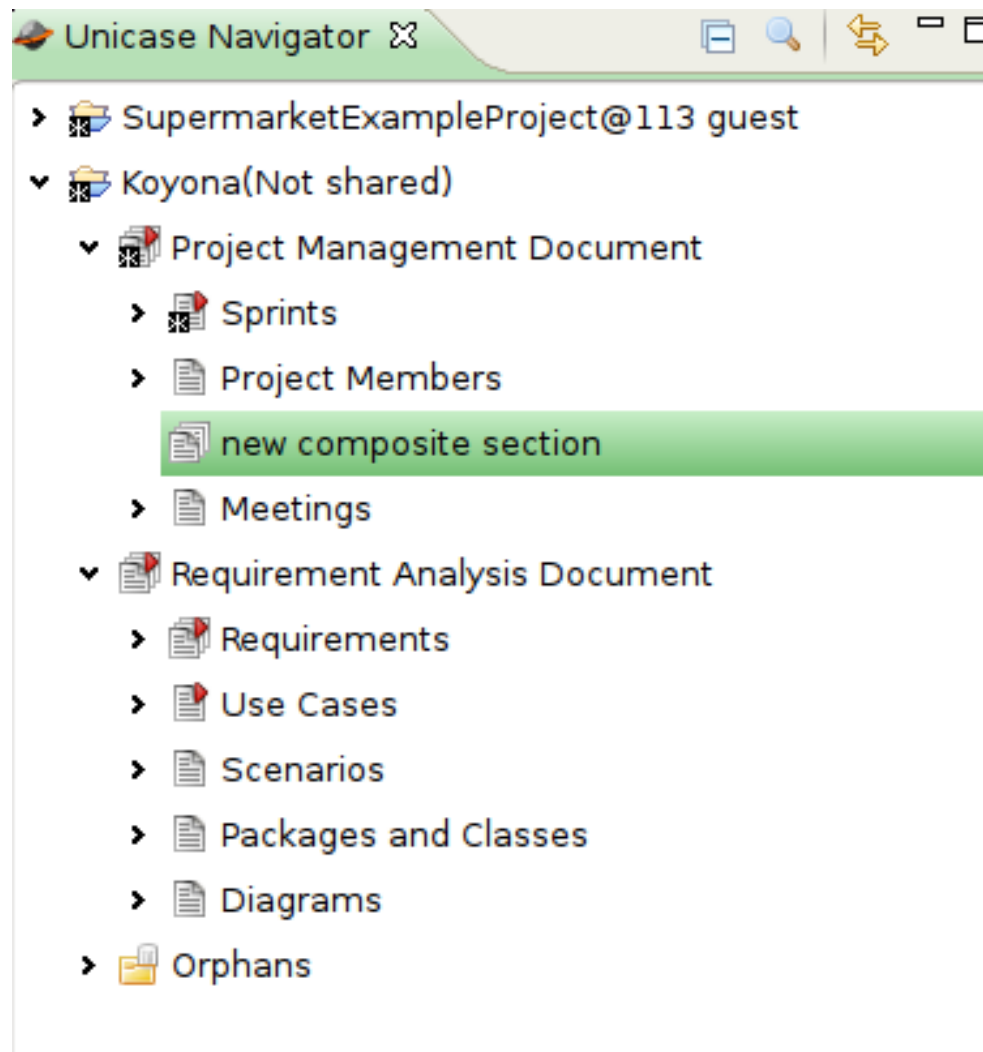
# Grupların proje üzerindeki hakları



# Kullanım

- Sunucu üstünde yeni bir proje oluşturuyoruz. Geliştiriciler sunucuya bağlanıp proje listesini alıyorlar. İlgili projede okuma/yazma gibi haklar tanımlandığını varsayıp o projeyi checkout (yerele kopyalama) etmeleri beklenir.
- Sunucudan çekilen projenin bir iskeleti olmalıdır. Bu iskeleti yönetici hazırlar. Geliştiriciler kendi geliştirdikleri alanları bu iskeletten bulup düzenlerler.
- İskelet hazırlama işi Leaf ve Composite denen kesimler tanımlanarak gerçekleştirilir. Composite(bileşke) kesimler composite ve leaf kesimleri içerebilir. Leaf kesimler ise “Model Element” denen nesneleri içerirler. Model Element bir diagram, aktör, hata, ya da bir görev tanımı olabilir. Bunlar ne anlama geliyor örneklerle görelim.

# Navigator



# Proje Üyesi Ekleme

- Project Management Document diye bir composite section ekleyip bunun altına Project Members diye bir leaf section ekleyip proje üyelerini buraya yazıyoruz. Bu üyeler toplarıya çağrılmadan tutun da belirli bir görevin bu kişiye atanmasına kadar bir çok yerde kullanılıyor olacak.

# ... Proje Üyesi Ekleme

- Kullanıcı ikonunun yanındaki siyah yıldız commit edilmediğini gösteriyor.

The screenshot shows a web application interface with a browser window. The browser has two tabs: 'Dashboard [Koyona]' and 'Project Members'. The 'Project Members' tab is active, showing a page titled 'Project Members [LeafSection]'. The page has a header bar with icons for trash, document, error, chat, star, and print. The main content area is divided into two columns. The left column has a 'Name' field with the value 'Project Members', a 'Description' field (empty), and a 'Comments' section showing '0 comments'. The right column has three expandable sections: 'Model Elements', 'Referenced Model Elements', and 'Annotations'. The 'Model Elements' section is expanded, showing three entries: 'mehmet', 'gokmen', and 'caglar', each with a user icon and a red 'X' mark. The 'Referenced Model Elements' and 'Annotations' sections are collapsed. At the bottom, there is a navigation bar with tabs: 'Standard View', 'Description', and 'Discussion'.

Dashboard [Koyona] Project Members Project Members X

Project Members [LeafSection]

Name Project Members

Description

Comments 0 comments

Model Elements

- mehmet X
- gokmen X
- caglar X

Referenced Model Elements

Annotations

Standard View Description Discussion

# Commit sonrası ve History

Project Members [LeafSection]

NameProject Members

Description

Model Elements

mehmet

gokmen

caglar

Referenced Model Elements

History for Koyona

> Local revision

> [HEAD,BASE] \*Version 2 [super @ 2011-06-20, 01:23] proje üyeleri eklendi.....

> Version 1 [super @ 2011-06-20, 01:10] ilk commit

> Version 0 [super @ 2011-06-20, 01:00] Creating project 'Koyona'

# Toplantı Yönetimi

Project Members

Project Management D

Meetings

TeamMeeting

Koyona Gereksinim Be

»2

Name

Koyona Gereksinim Belirleme Toplantısı

▼ Participants

caglar

✖

gokmen

✖

mehmet

✖

▼ Sections

Modüler Yapı

✖

▼ Annotations

▼ Attachments

Description

Comments

0 comments

Location

CZ34

Starttime

20.06.2011 01:37

✖

Endtime

20.06.2011 01:45

✖

Facilitator

mehmet

✖

Minutetaker

mehmet

✖

Timekeeper

mehmet

✖

Identified Issues Section

Modüler Yapı







✖























Identified Work Items Section

(Not Set)

# Issue Ekleme

Modüler yapı tercih edilmeli mi? ✖

Modüler yapı tercih edilmeli mi? [Issue]      

Name	Modüler yapı tercih edilmeli mi?	▼ Participants  
Description	<div>Modüler uygulama geliřtirmenin avantajları nelerdir? Bunu tercih etmeli miyiz?</div>	▼ Predecessors  
Comments	<a href="#">0 comments</a>	▼ Successors  
Activity	System Design	▼ Annotated Model Elements  
Containing Workpackage	 <a href="#">İlerleme Raporu mu bu sprint n[...]</a> ✖  	▼ Annotations  
Assignee	 <a href="#">mehmet</a> ✖  	▼ Attachments  
Reviewer	 <a href="#">gokmen</a> ✖  	
Due Date	(Not Set)	
Priority	2 	



# Sprint

*“Sprint; agile olarak geliştirilen sistemlerde belirli özellikleri içeren program parçalarının, yapılan işlerin demosu. Bir çeşit müşteri veya proje yönetimi gözden geçirmesi”*

Name ilerleme Raporu mu bu sprint ne demek

Description

Comments [0 comments](#)

Containing Workpackage (Not Set)

Assignee [mehmet](#)

Reviewer [gokmen](#)

Due Date (Not Set)

Priority -10

Resolved ☐

Estimate 0

▼ Participants

▼ Predecessors

▼ Successors

▼ Contained Work Items

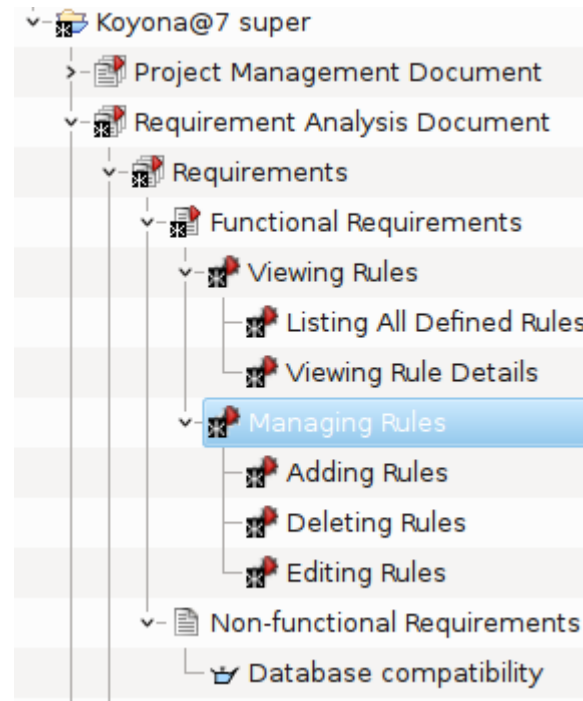
[Modüler yapı tercih edilmeli mi?](#)

▼ Annotated Model Elements


▼ Annotations

▼ Attachments

# İşlevsel Gereksinim Ekleme



# ... İşlevsel Gereksinim Ekleme



## Managing Rules [FunctionalRequirement]

Name	Managing Rules	
Description	<div></div>	
Comments	<a href="#">0 comments</a>	
Refined Requirement	(Not Set)	
Stakeholder	(Not Set)	
Priority	0	
Reviewed	<input type="checkbox"/>	
Cost	0	

▼ Use Cases

▼ Scenarios

▼ Refining Requirements

Adding Rules

Deleting Rules

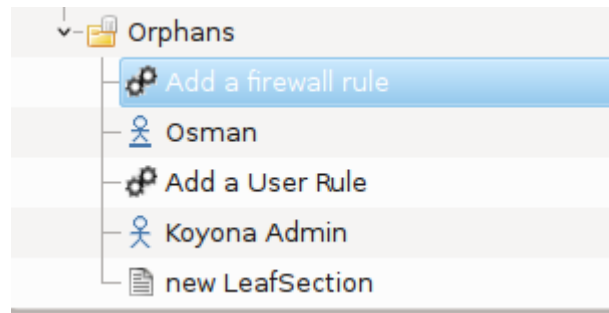
Editing Rules

▼ Annotations


▼ Attachments







# Orphans

Herhangi bir modelden silinen nesne sistemden doğrudan kaldırılmıyor; orphans dizini altında saklanıyor. Bunları tekrardan modellere dahil etmek mümkün.



# Use case

 **List User Rules [UseCase]**




Name





List User Rules

Description



Comments

 [0 comments](#)



Initiating Actor

 [Koyona Admin](#)   



Realized User Task

(Not Set)  

▼ Participating Actors



 


▼ Included Use Cases



▼ Extended Use Cases



▼ Scenarios



 

 [Listing User Rules](#)



▼ Functional Requirements



 [Viewing Rules](#) 



 [Listing All Defined Rules](#) 



▼ Non Functional Requirements


 

▼ Identified Classes

 [UserRule](#) 

 [Rule](#) 

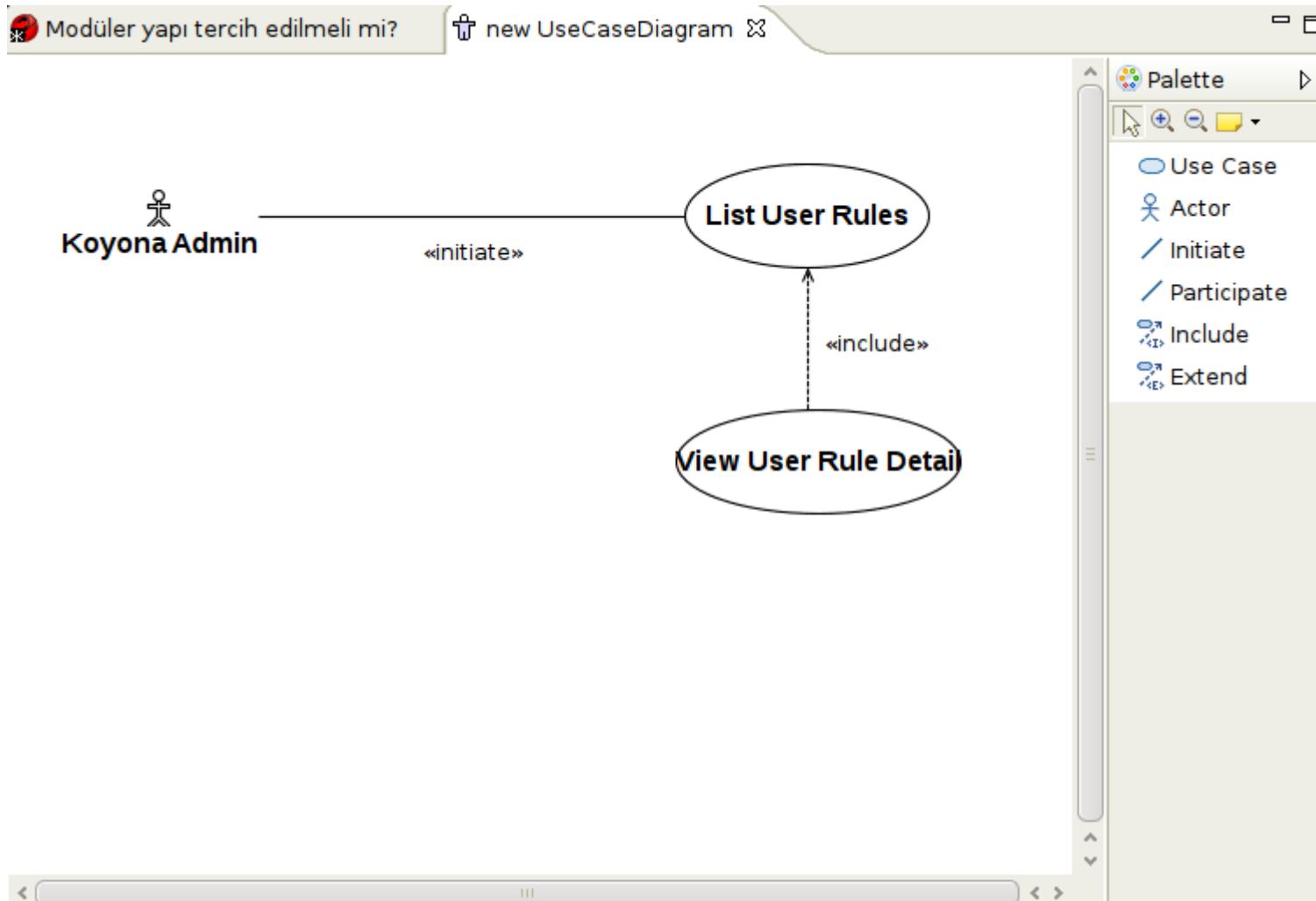
 [RemoteReader](#)

# Use case için senaryo

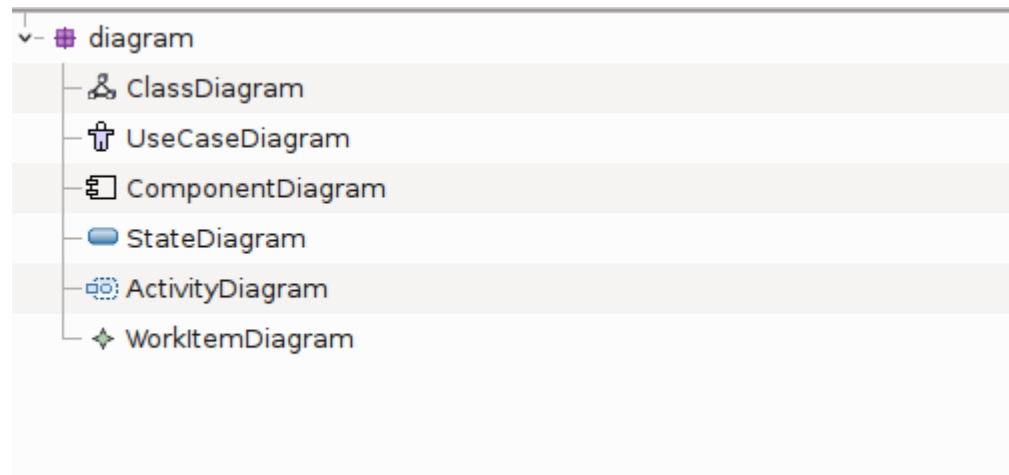
[illegible]

# Usecase diagram

Tanımlanan nesneleri navigatordan sayfaya sürükleyerek diagram oluşturabiliyoruz.

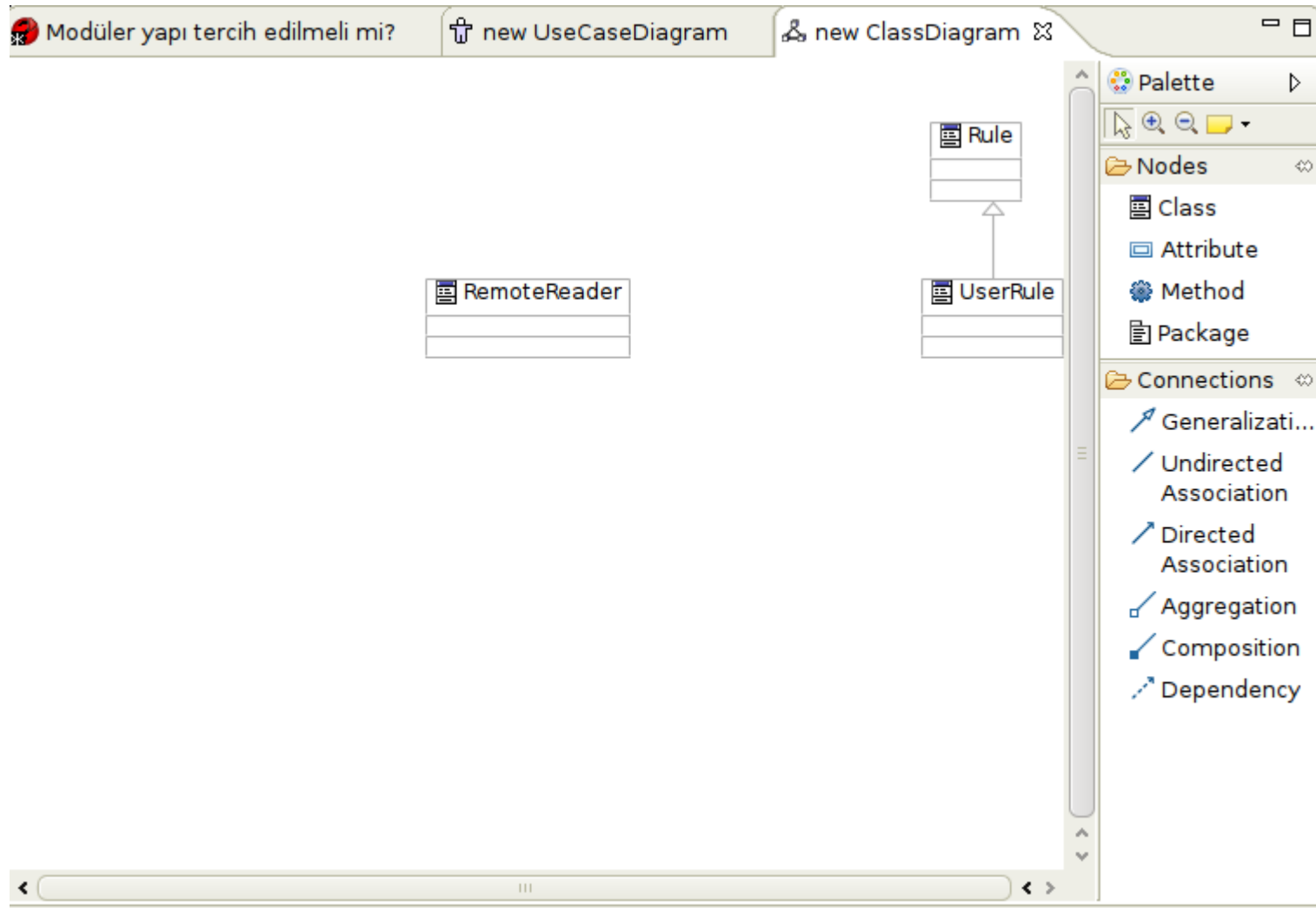


# Diyagramlar

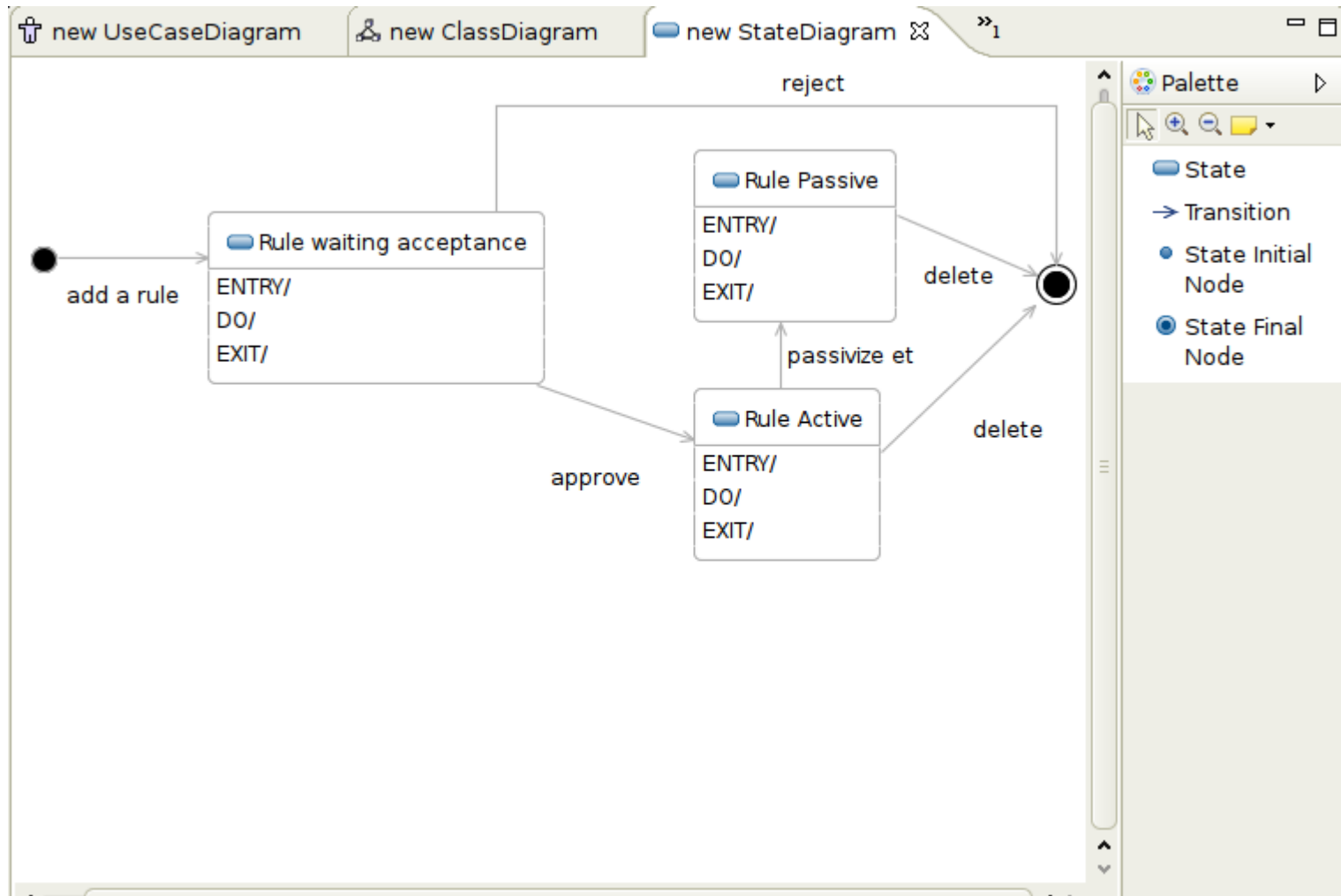




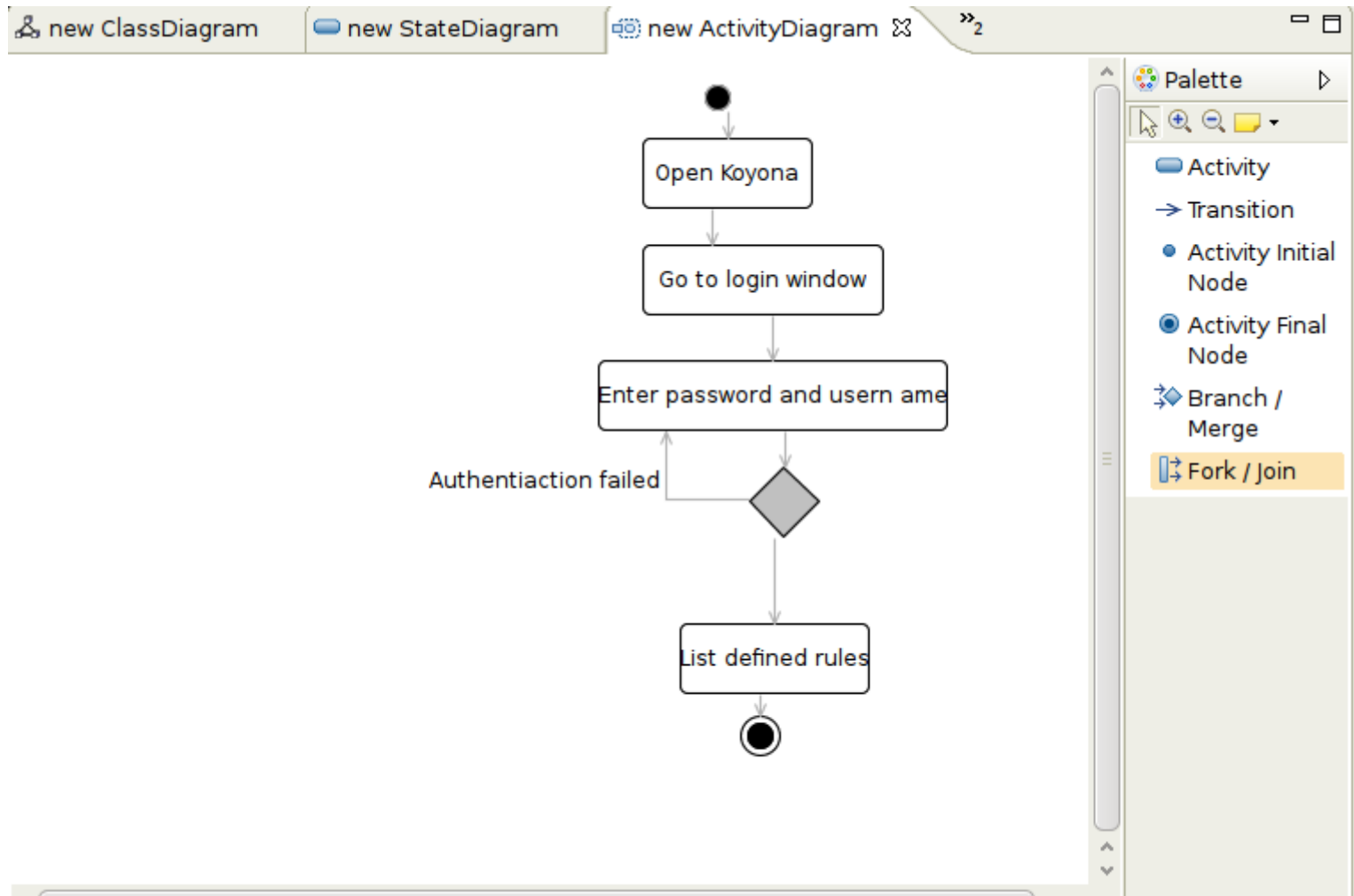
# Class diagram



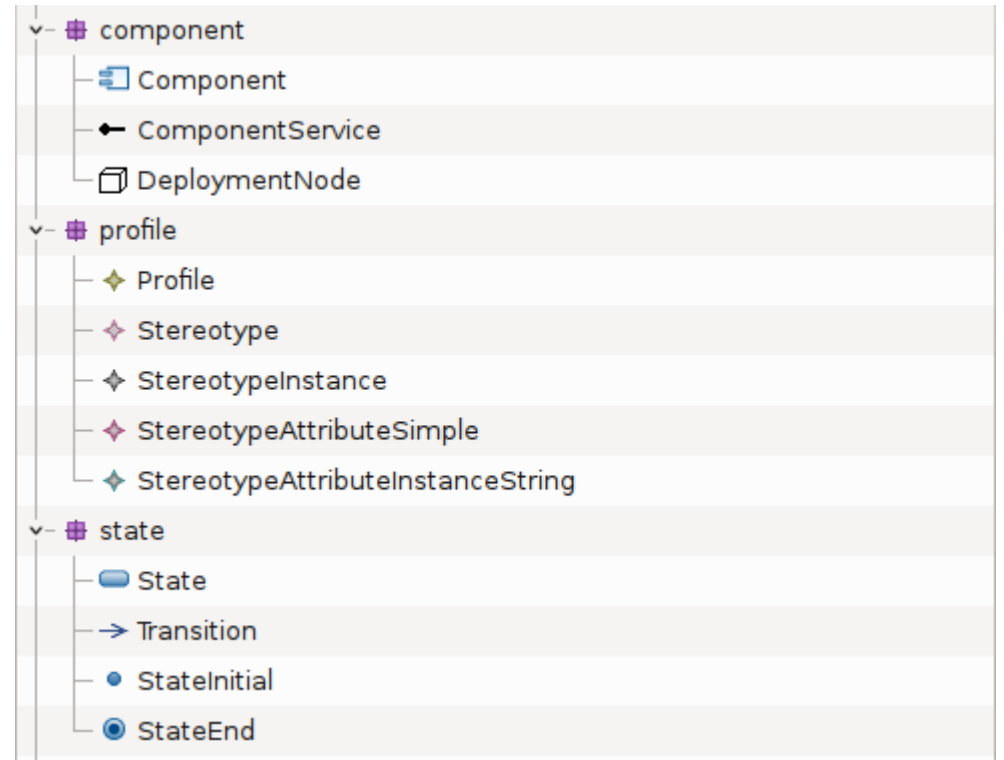
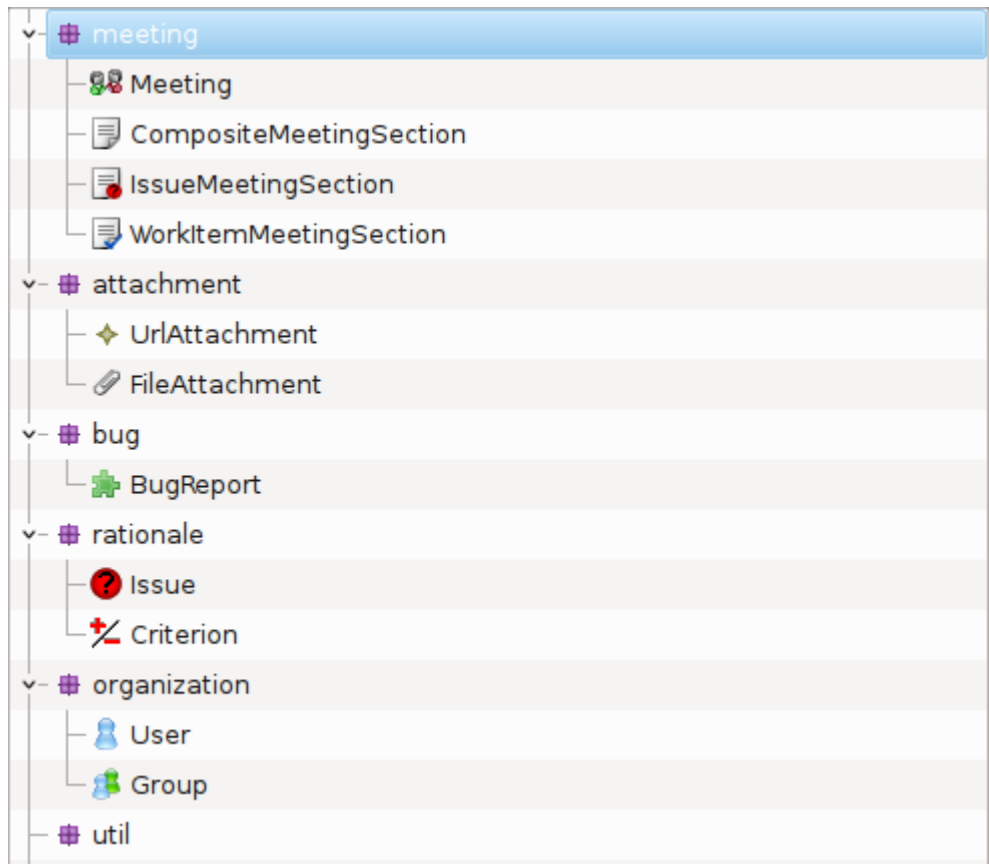
# State diagram



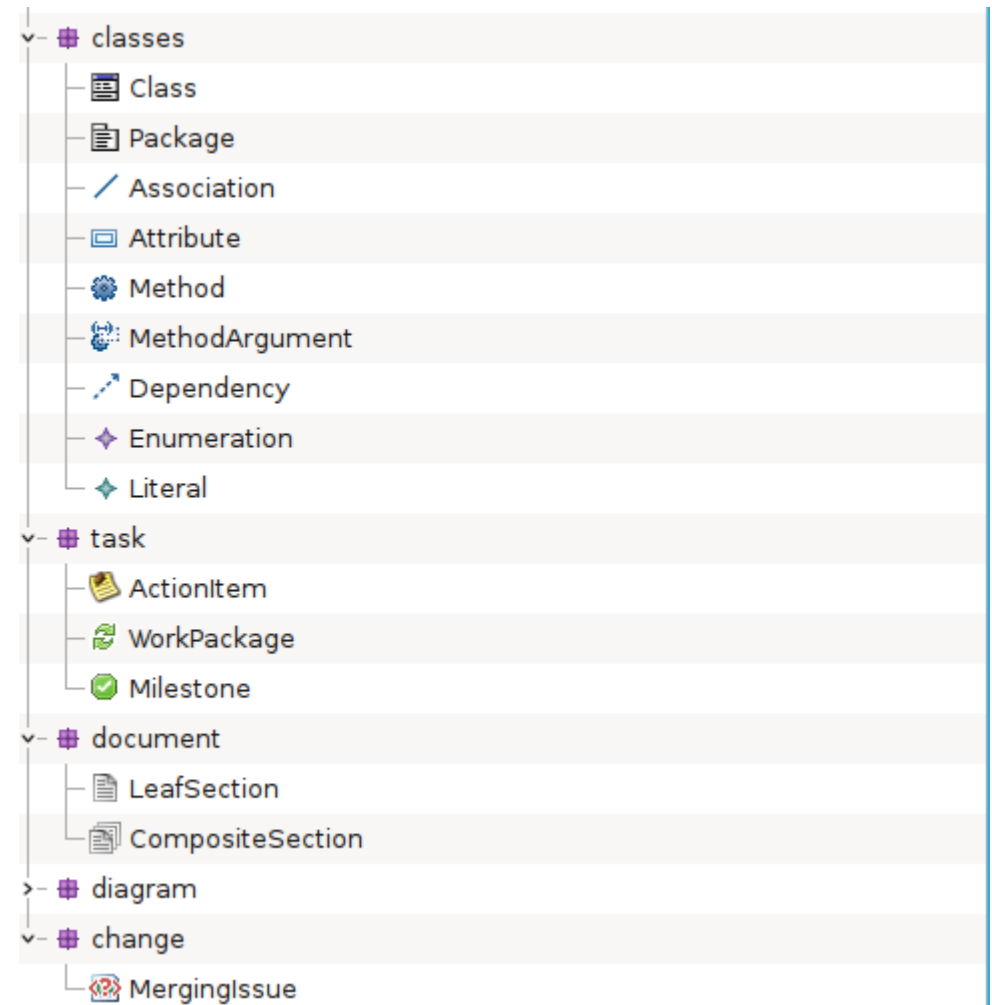
# Activity diagram



# Model Elements



# ... Model Elements



# Sequence & Collaboration

- Bu diagramlar yok!!!
- *“Deployment diyagramı da yok. Ama bir şekilde component diyagramı ile birleştirebiliriz gibi görünüyor. (Component diyagramının içinde deployment node'u verilmiş)”*