

## Süsteemianalüüs, nõuete püstitamine

---

Toomas Saarsen

Ahto Truu

Are we building the software right?

VS

Are we building the right software?

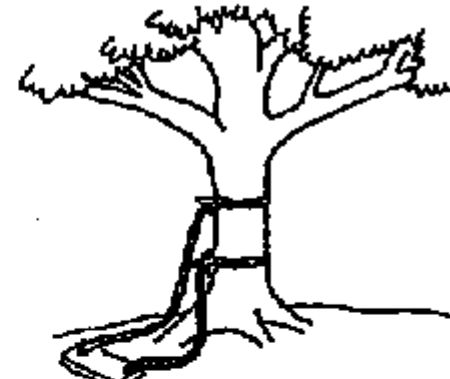
# Hoiatav näide



As proposed by the project sponsor.



As specified in the project request.



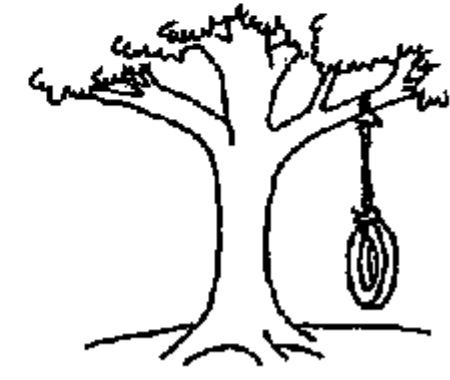
As designed by the senior analyst.



As produced by the programmers.



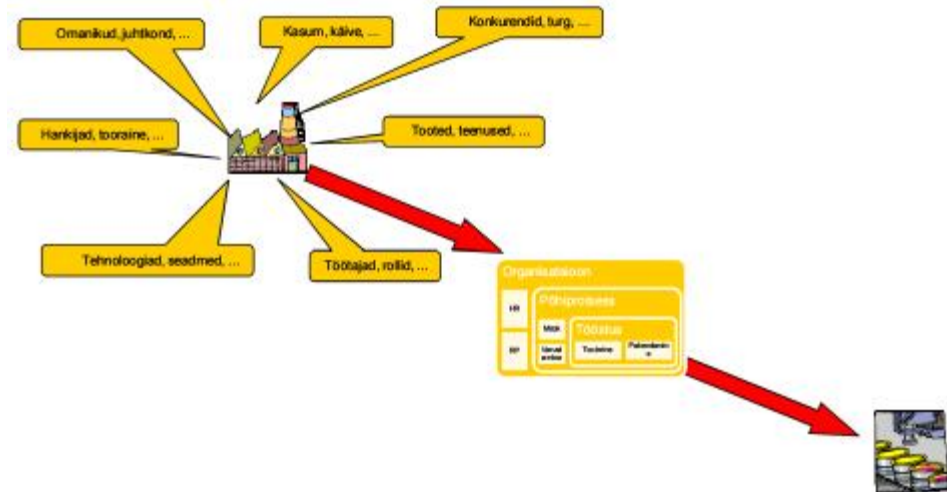
As installed at the user's site.



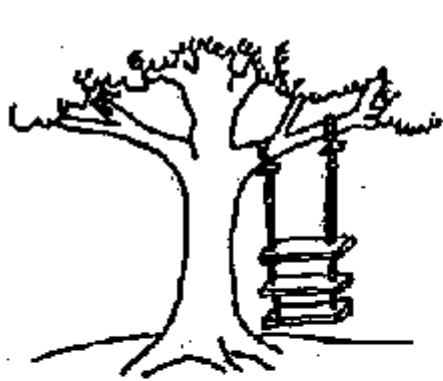
What the user really wanted.

- Kes on süsteemi kasutajad?
- Milleks neil seda süsteemi vaja on?
- Liidesed väliste süsteemidega?

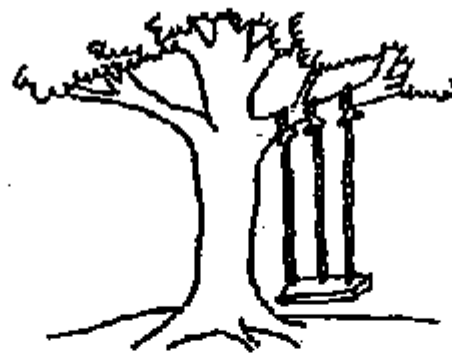
- *Unified Modeling Language*
- Formaalne graafiline keel
- UML 1: 1997  
Ivar Jacobson,  
Grady Booch,  
Jim Rumbaugh
- UML 2: 2005  
OMG standard



# Hoiatav näide



As proposed by the project sponsor.



As specified in the project request.



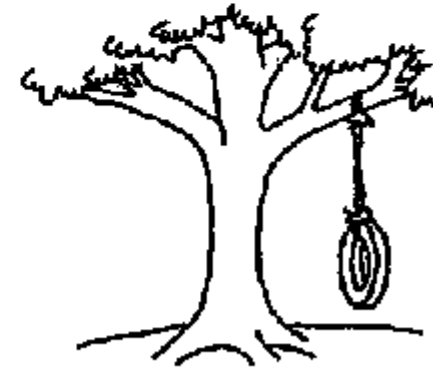
As designed by the senior analyst.



As produced by the programmers.



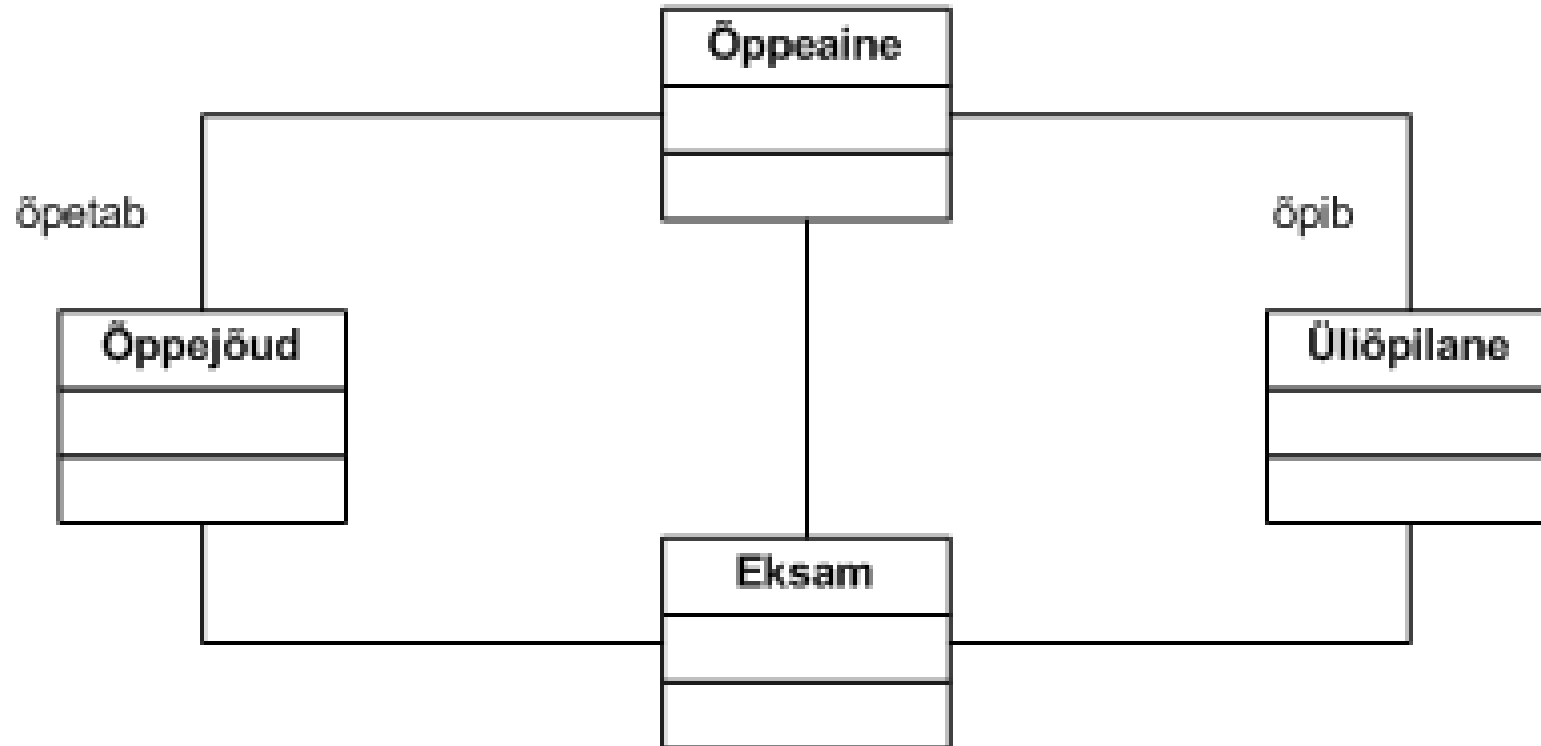
As installed at the user's site.



What the user really wanted.

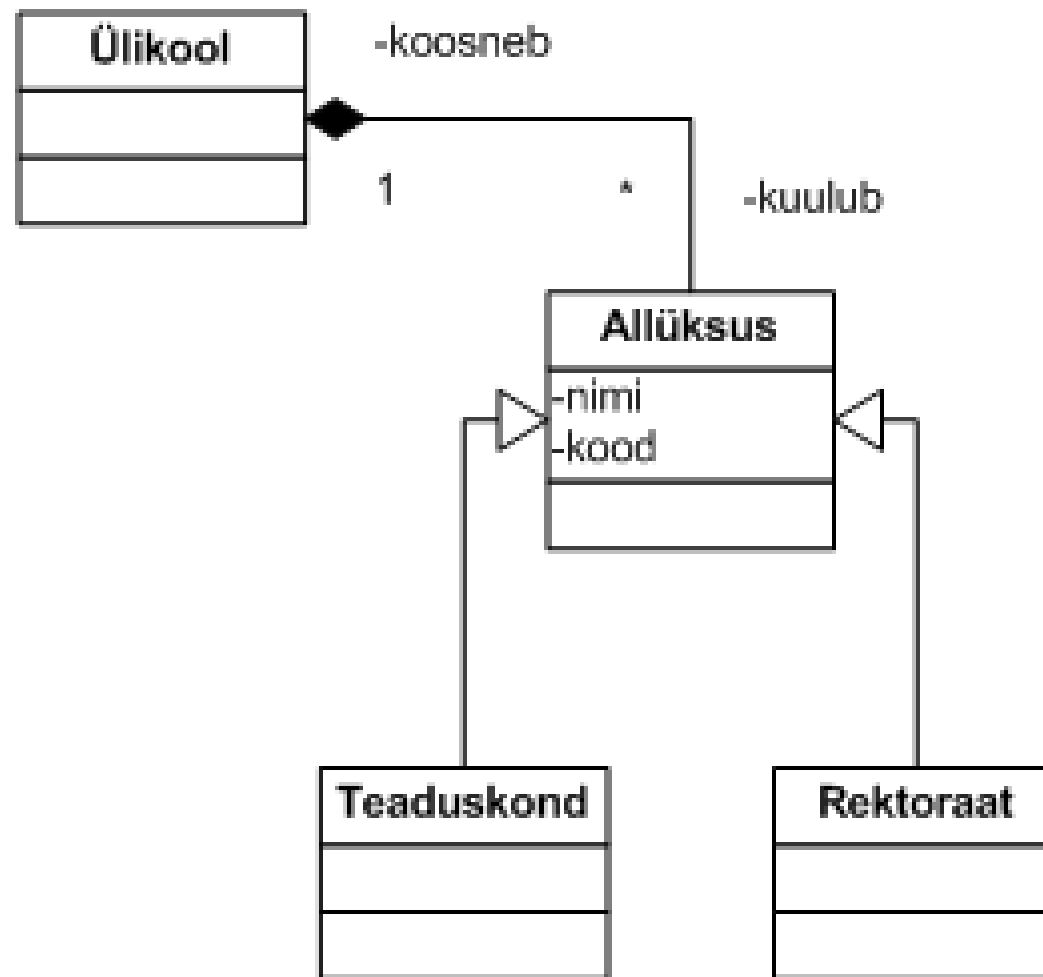
- Valdkonna mudel (*domain model*)
  - Millega me üldse tegeleme?
- Klass (*class*)
  - Sarnaste objektide kogum
- Seos (*relationship*)
  - Objektide omavahelised suhted

# Näide





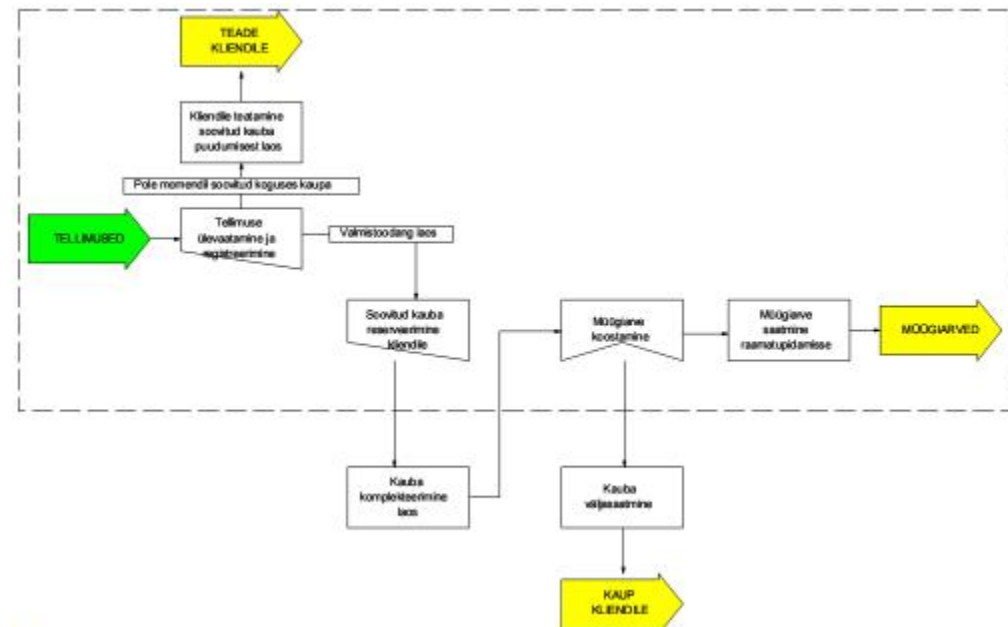
# Näide



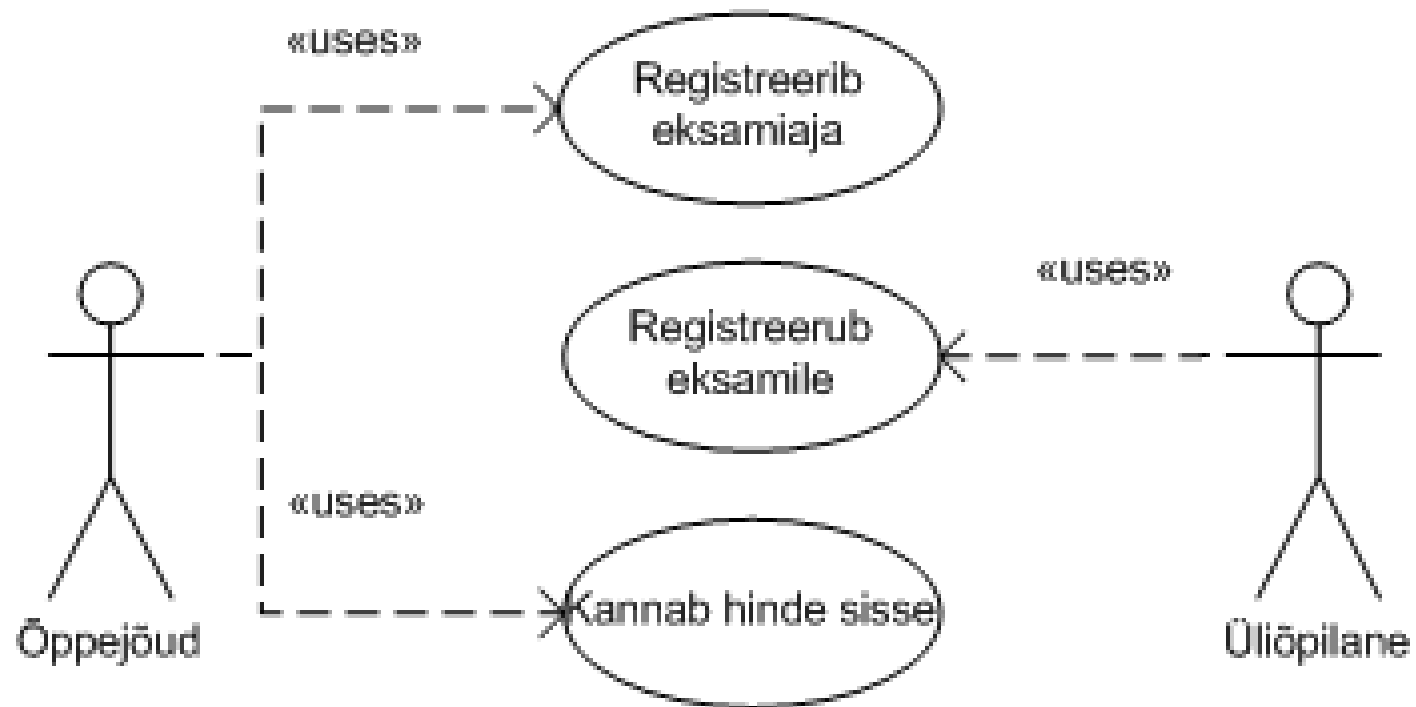
- Modelleeri ainult valdkonna mõisted
- Nimeta asju kasutaja terminitega
- Ära hakka (veel) tarkvara või andmebaasi projekteerima
- Hoidu meetodite spetsifitseerimisest
- Ära atribuutide ja seostega üle pinguta

## Kasutusjuhud, kasutuslood

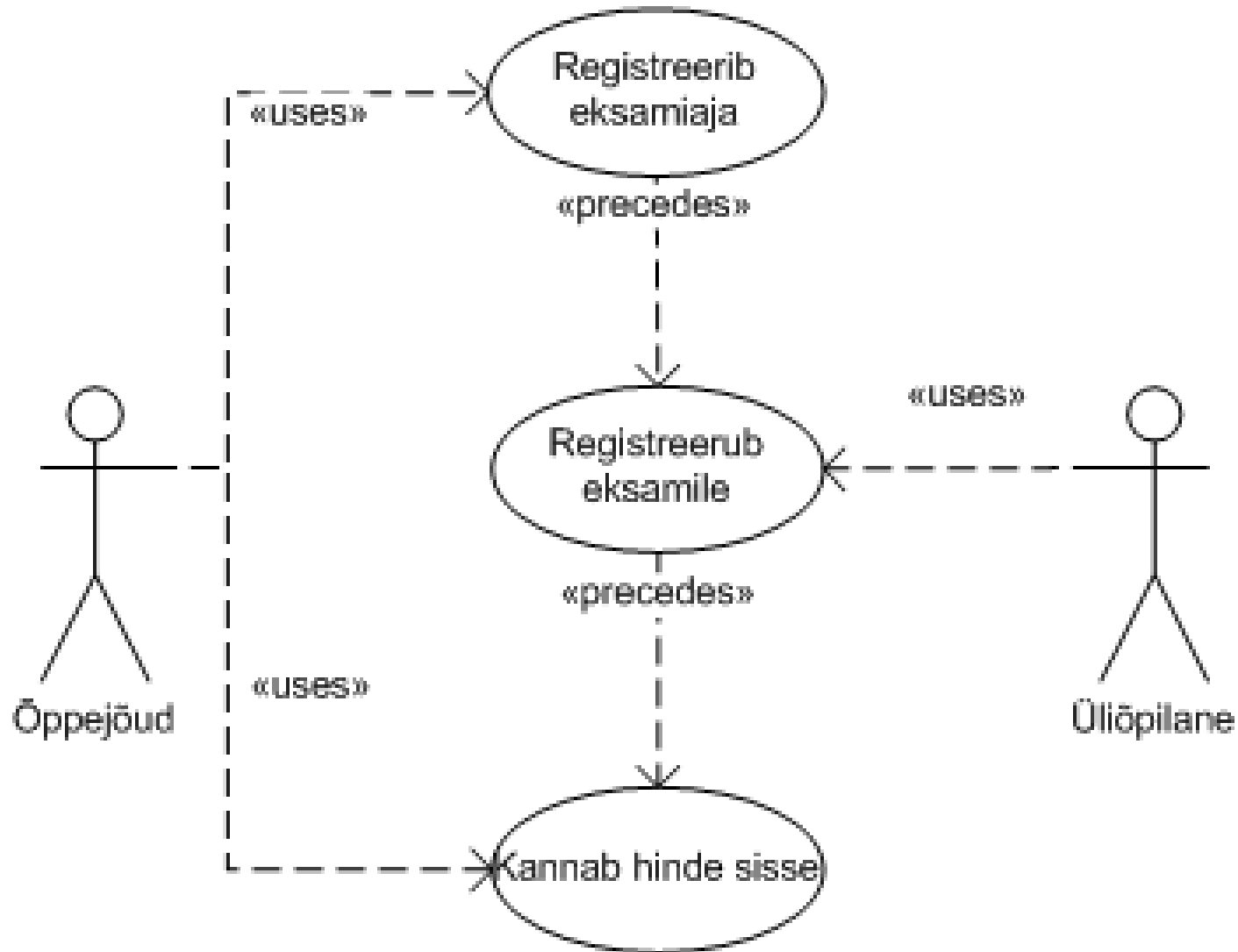
- Kasutusjuht (*use case*)
- Kasutuslugu (*use case story*)
- Kasutaja (actor)



# Näide



# Näide



- Keskendu sellele, mis toimub kasutusjuhu sees, mitte enne või pärast
- Keskendu kasutaja vaatenurgale, kuid ära unusta ka süsteemi käitumist
- Kirjelda ka alternatiivsed stsenaariumid
- Kirjelda ka negatiivsed stsenaariumid
- Väldi umbisikulist tegumoodi
- Põhistsenaarium  $7 \pm 4$  sammu

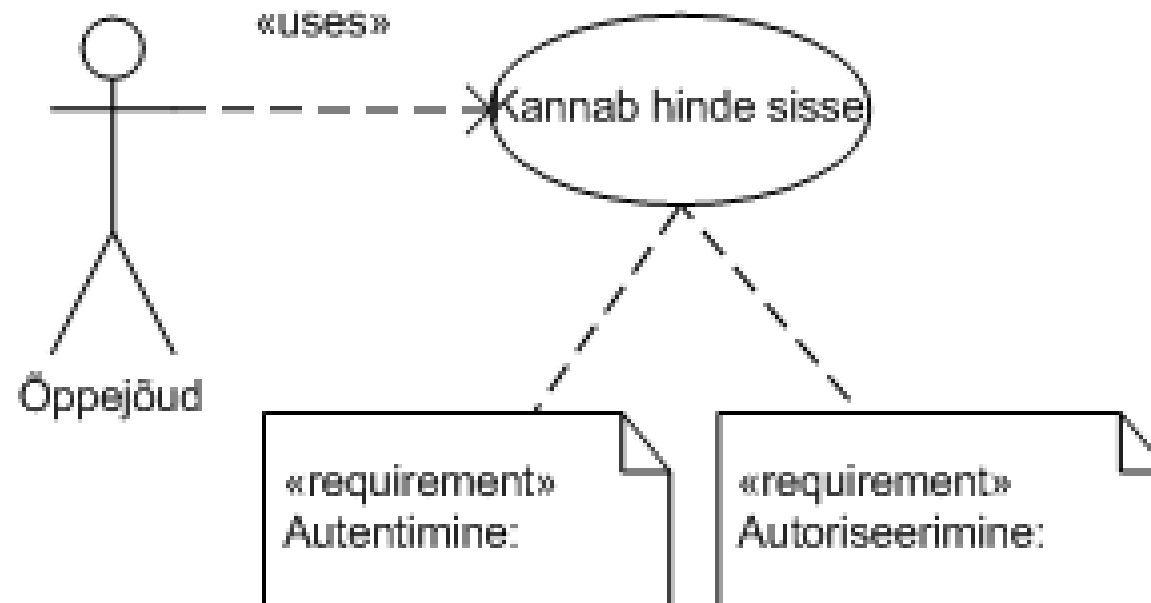
- Funktsionaalne nõue  
(*functional requirement*)
  - Mida tarkvara peab tegema?
- Mittefunktsionaalne nõue  
(*non-functional requirement*)
  - Kuidas ta seda peab tegema?
  - Mida ta ei tohi teha?

- Kasutatavus (*usability*)
- Töökindlus (*reliability*)
  - Käideldavus (*availability*)
  - Tõrkekindlus (*fault tolerance*)
- Jõudlus (*performance*)
- Hallatavus (*supportability*)
  - Konfigureeritavus
  - Dokumenteeritus



- UMLis puudub element “nõue”
- Nõuded tekstina kasutusloos sees
- Nõuded märkustena (*note*)  
kasutusloos küljes
- Nõuded klassidena kasutusloos  
küljes

# Näide



Kui juba nõudeid modelleerida, siis

- Iga nõue peab olema seotud vähemalt ühe kasutusjuhuga
- Iga nõudega peab olema seotud vähemalt üks klass
- Iga nõudega peab olema seotud vähemalt üks test

- UML distilled  
Martin Fowler  
Addison-Wesley, 2003
- Use Case Driven Object Modeling with UML  
Doug Rosenberg, Kendall Scott  
Addison-Wesley, 1999
- Writing Effective Use Cases  
Alistair Cockburn  
Addison-Wesley, 2000
- UML-keele sõnastik  
Mati Littover, Lauri Lubi  
<http://www.cc.ioc.ee/uml/>