TENIS-PISTEN ERRESERBEN KUDEAKETA

Tenis-klub batek bere bost pistak erreserbatzeko aplikazio bat garatu nahi du. Klubeko bazkideek erreserbatu ahal izango dituzte pistak. Erreserbei buruz honako ezaugarriak bete beharko ditu sistemak:

- Pistak orduka erreserbatuko dira.
- Erreserba bertan behera utzi daiteke, baldin eta erreserbaren egun berean egiten ez bada.
- Gehienez ere hilabete bat aldez aurretik egingo dira erreserbak.
- Lau salneurri daude: T1, asteburuetarako eta laneguneko gau-orduetarako; T2, beste orduetarako; T3, erreserba erabiltzen ez denerako; eta T4, erreserba bertan behera uzten denerako.

Gainera, hilero, bazkide bakoitzari faktura bat bidaliko zaio. Bertan pisten erabilera eta bazkidearen izena eta helbidea agertuko dira (sistemak fakturak inprimatu egingo ditu: ez bidali)

Lor itzazu aurreko informazio sistemari dagozkion Erabilpen Kasuen Eredua (gertaeren fluxuak barne) eta Domeinuaren Eredua

OHARRAK:

- *Erreserbak eta fakturazioa kudeatzen dituzten erabilpen kasuak **besterik ez** dira egin behar. Bazkideak kudeatzen dituztenak **ez**. Erreserbak eta fakturazioa **ondo egiteko beste erabilpen kasuren bat sartu ahal izango da** (enuntziatuan agerian ez agertu arren)*
- *Deskribatu erabilpen kasu horiei dagozkien gertaeren fluxu **egokiak**.*
- *Gertaeren fluxu horien arabera, domeinuaren ereduan agertuko beharko dute beharrezko klase, atributu eta asoziazioek.*

Software Ingeniaritza
Sekuentzia Diagramak Diseinatzeko Dekalogoa

1. Hiru mailako aplikazio batean gutxienez hurrengo klaseko objektuak agertu behar dira: Interfaze grafikoa, negozio logikako kudeatzailea, datu atzipen kudeatzailea, datu-base kudeatzailea eta domeinuzko objekturen bat.
2. Interfaze grafikoren helburu nagusiena erabiltzailetik parametroak jaso, negozio logikari deitu emaitzak bistaratu da.
3. Interfaze grafikoa hurrengo objektuekin komunikatu daiteke soilik: negozio logikako kudeatzailearekin eta domeinuzko objektueen GETTER metodoekin (objektuen egoera aldatzen ez duten metodoekin)
4. Negozio logikako kudeatzailean interfaze grafikoek behar dituzten eragiketa guztiak kokatu beharko dira.
5. Negozio logikako kudeatzailea, datu atzipen kudeatzailea, eta domeinuzko objektuekin (GETTER metodoekin) komunikatu daiteke soilik (**Inoiz ere interfaze grafikoarekin**).
6. Hasiera batean domeinuzko objektu guztiak datu basean gordeta daude. Objektu bateri deitu nahi bada, lehendabizi memoria nagusira ekarri beharko da (erreferentzia bat lortuz).
7. Datu atzipen kudeatzaileak beharrezko eragiketak eduki behar ditu datu basean dauden domeinuzko objektu guztiak memoria nagusira ekartzeko.
8. Datu atzipen kudeatzailea, datu base kudeatzaile (hau da Db4o) eta domeinuzko objektuekin komunikatu daiteke soilik (**Inoiz ere UI eta NL-ko kudeatzaileekin**)
9. Negozio logikako eragiketaren bat pertsistentzia behar badu (Adb: landetxea sortu) datu atzipen kudeatzailearer bitartez gauzatuko du. Hau da, datu atzipen mailan, método bat inplementatu beharko da funtzionalitate horrekin.
10. Domeinuzko objektuak beste negozio logika klaseko objektuekin komunikatu daiteke soilik (Ez UI, DB eta NL kudeatzaileekin).

"Tarta"

Proiektu bateratua

5' gehienez
Igandeerako (24:00)

① Zer da?

- Softwarea garatzeko metodo bat, iteratibo eta inkrementala
-

---

② Zer da iteratibo eta inkrementala izatea?

- 4 fase: Hasiera, Elaborazioa, Eraikuntza eta Trantsizioa.
- Fase hauek iterazioetan, iterazio bakoitzean lau fase hauek betetzen dira.

- Adib: Tarta

---

③ Zelako ondorioak ditu metodologia honek?

> Lana errazten du

> Egitekoak banatzen dira

> Ez gara inoiz esku-hutsik gelditzen

> Arazo bat izanez gero, arazo horri jartzen zaio arreta

Fernando arrazketarra

Irakasgaia: Software Ingeniaritza (http://goo.gl/gNbjL)

Irakaslea: Jon Iturrioz (email: jon.iturrioz@ehu.es)

Tutoretzak: Astelehena:11:30-12:30 eta 15:00-17:00, osteguna:10:30-13:30

Laborategia: E4 (gela 1.1)

Aurkibidea

1. GAIA: Eskakizunen Bilketa UML erabiliz

1.1.- Erabilpen kasuen diagramak
1.2.- Domeinuaren diagramak

2. GAIA: Deseinua

2.1.- Sekuentzia diagramak UML erabiliz
2.2.- GRASP Arduradun patroiak
2.3.- Maila anitzeko Software arkitekturak

3. GAIA: Inplementazioa JAVA lengoaian

3.1.-Interfaze grafikoak: SWING/AWT
3.2.- Objektuen persistentzia: db4o
3.3.- Konputazio banatua: Web Zerbitzuak

Oinarrizko bibliografia

- **UML, Prozesu bateratua eta GRASP patroiak**
  UML y Patrones
  Craig Larman. Prentice Hall, 2002
- **Use case driven modelling with UML**
  Doug Rosenberg and Math Stephens
  Apress, 2007
- **Java programazio lengoaia**
  Piensa en Java
  Bruce Eckel. Prentice Hall,2007 (4. edizioa)
- **Java online tutorial**
  http://www.javahispano.org/contenidos/es/java_basico_con_ejemplos/
- **Software ingeniaritzari buruzko liburu orokorra**
  Ingeniería del Software. Un enfoque práctico.
  Roger S. Pressman. MacGraw-Hill, 2001. 5ª Edición.
- **Objektu Zuzendutako Diseinua**
  Construcción de Software Orientado a Objetos.
  Bertrand Meyer. Prentice-Hall. 1998.

Software Ingeniaritza
Informatika Ingeniaritzako Gradua
Informazio Orokorra

Department of Computer Languages and Systems

EBALUAZIO JARRAITUA: Parte hartze aktibo eta sistematikoa eskatzen du. Ebaluazio Jarraitua aukeratuz ikasleak gutxienez hiru konpromiso hartuko ditu: eskola eta laborategietara etorri eta bertan parte hartzea, Proiektua garatuko duen talde batean integratzea, eta proposatutako jarduerak (ariketak, lanak, praktikak, azalpenak...) burutzea. Ebaluazio Jarraitua ikastaroaren hasieran aukeratu ahal izango da, eta behin betiko bihurtuko da adieraziko diren epeetan (11. astearen inguruan), irakasleak ikaslearen errendimendua egiaztatu ondoren.

Hauek dira ebaluazio jarraiturako kontuan hartuko diren ekintzak:

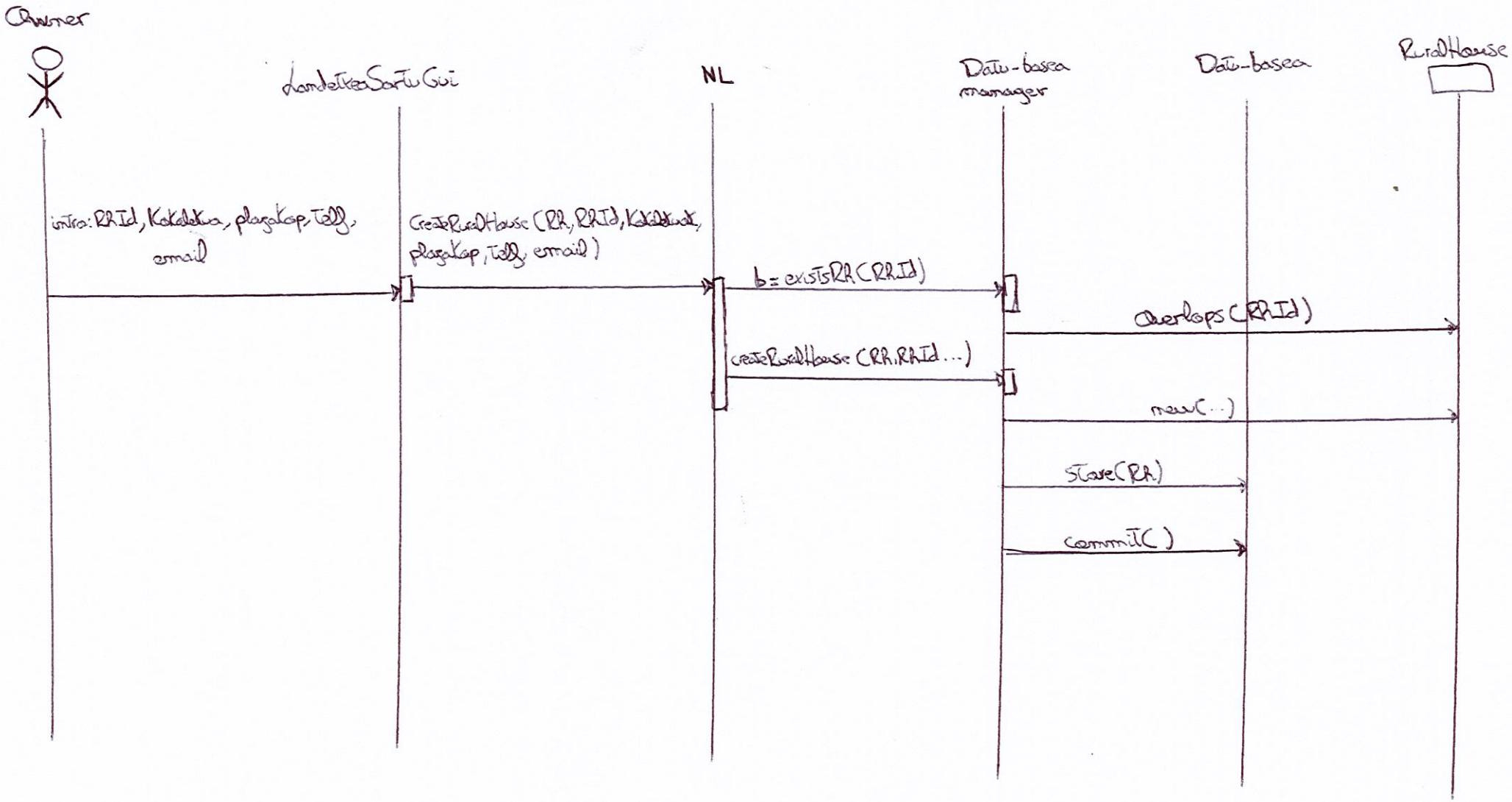
1) Proiektua: %75

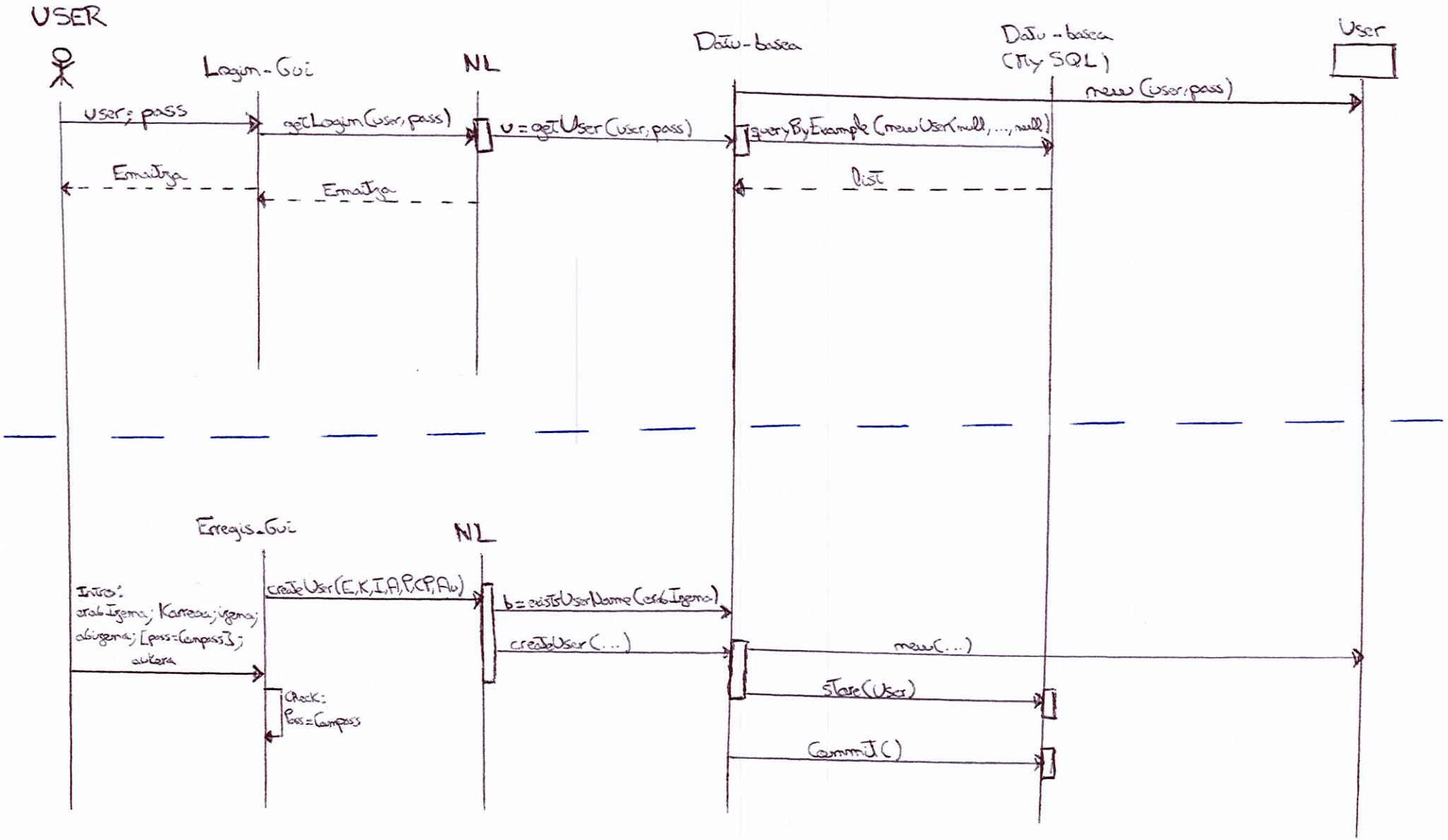
2) Beste ekintza osagarriak (laborategiak barne): %25

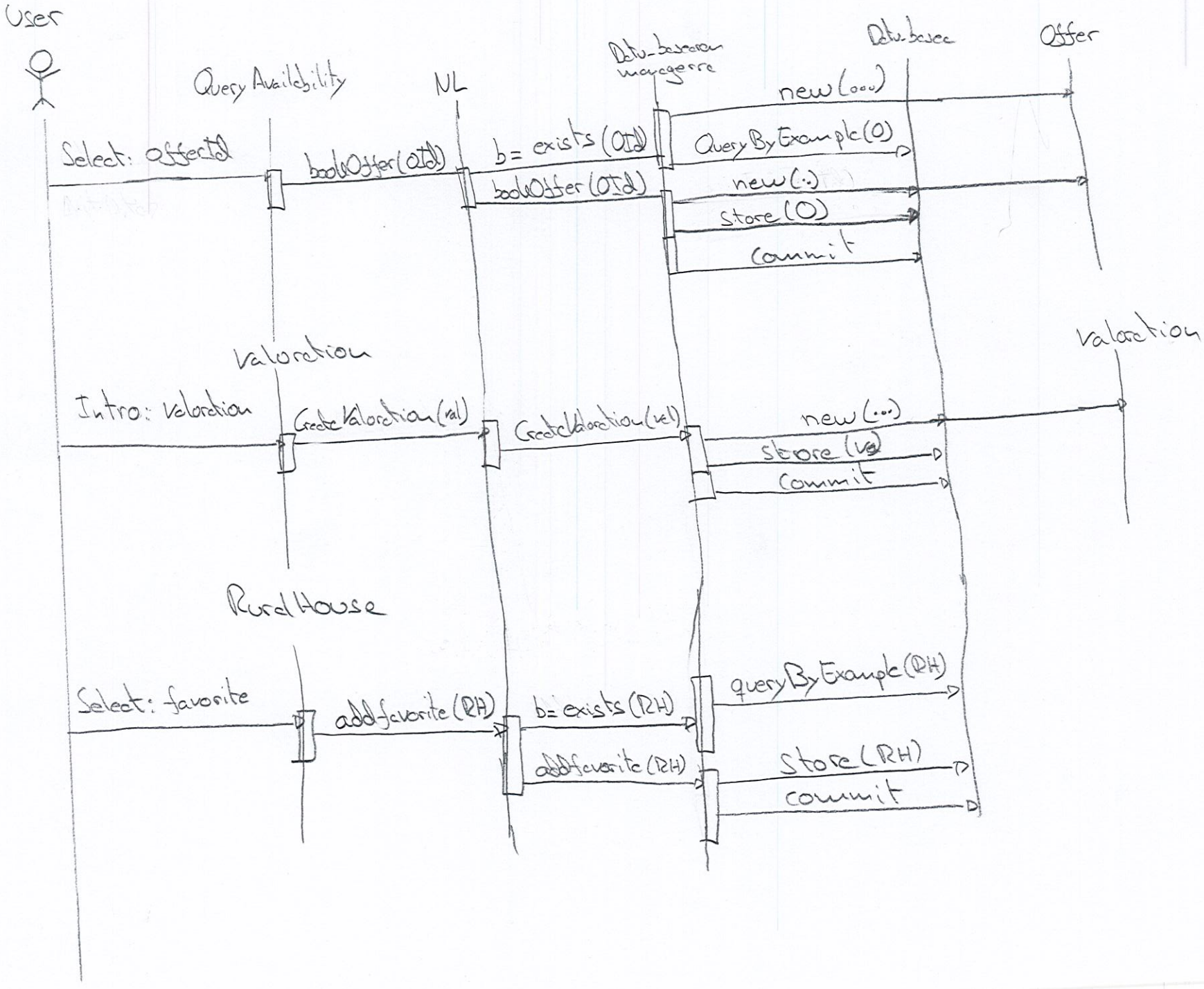
Proiektua hiru iteraziotan gauzatuko da, horietako bakoitzean garapenaren fase guztiak burutuz: eskakizunen deskribapena, diseinua eta software- arkitektura, eta inplementazioa. Hori dela eta 1., 2. eta 3. gaiak ez dira bata bestearen ondoren ikusiko, modu espiralean baizik. **Hirugarren iterazioan proiektuaren defentsa globala eskatuko da, galdetegi labur batekin batera.**

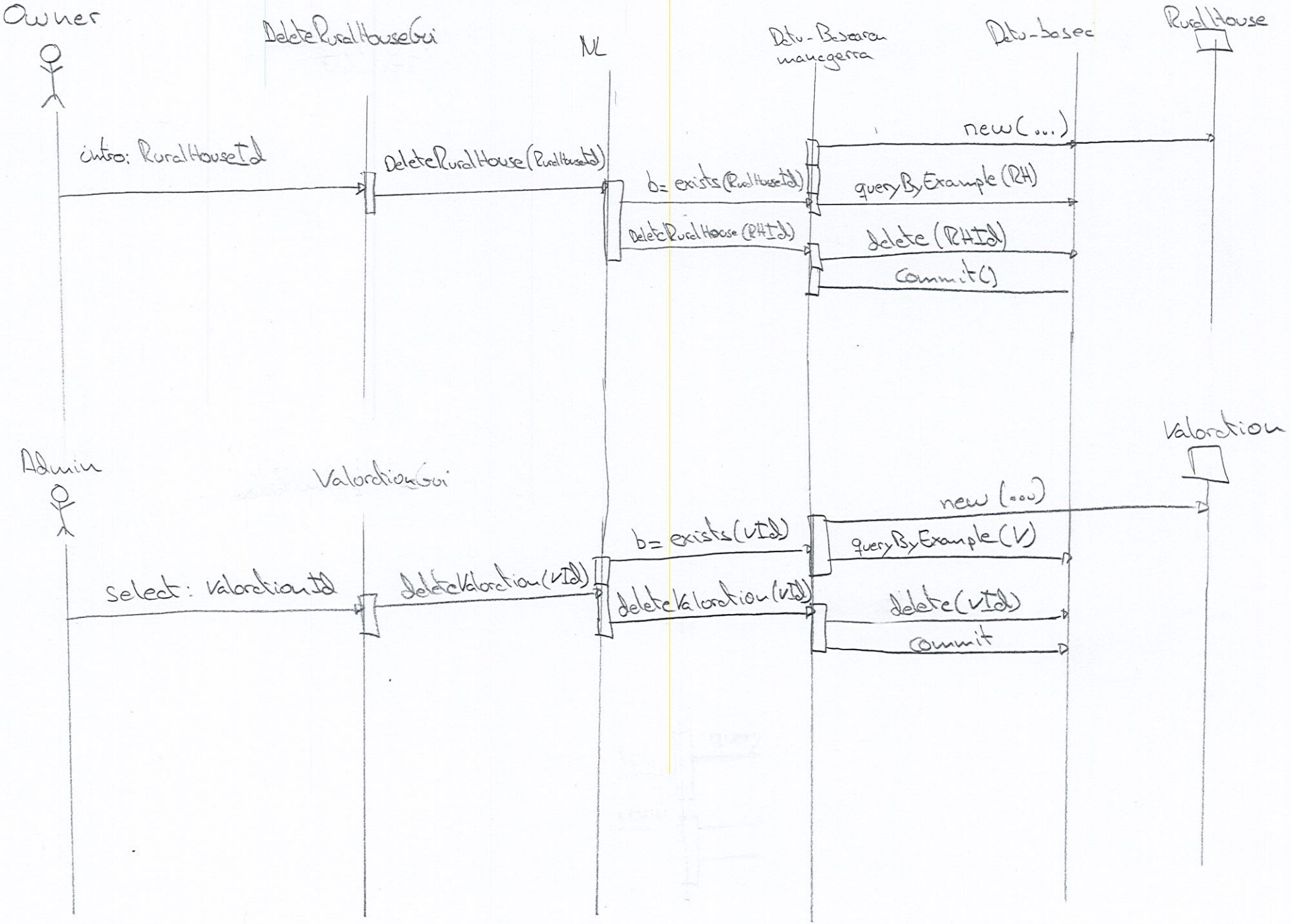
EBALUAZIO GLOBALA: Ebaluazio Jarraitua egin nahi edota ahal ez duten ikasleentzat aplikagarria da. Hala nola Proiektu bat, agian desberdina, burutu eta defendatzea derrigorrezkoa izango da ere bai. Ikasturtearen bukaeran ikasgaiaren edukien %100a barneratuko duen Azterketa Globala ospatuko da, eta azken nota Proiektua eta Azterketaren batezbestekoa izango da. **Ebaluazio Globalean parte hartzeko irakaslearekin zita adostu behar da Martxoak 22 aurretik.**

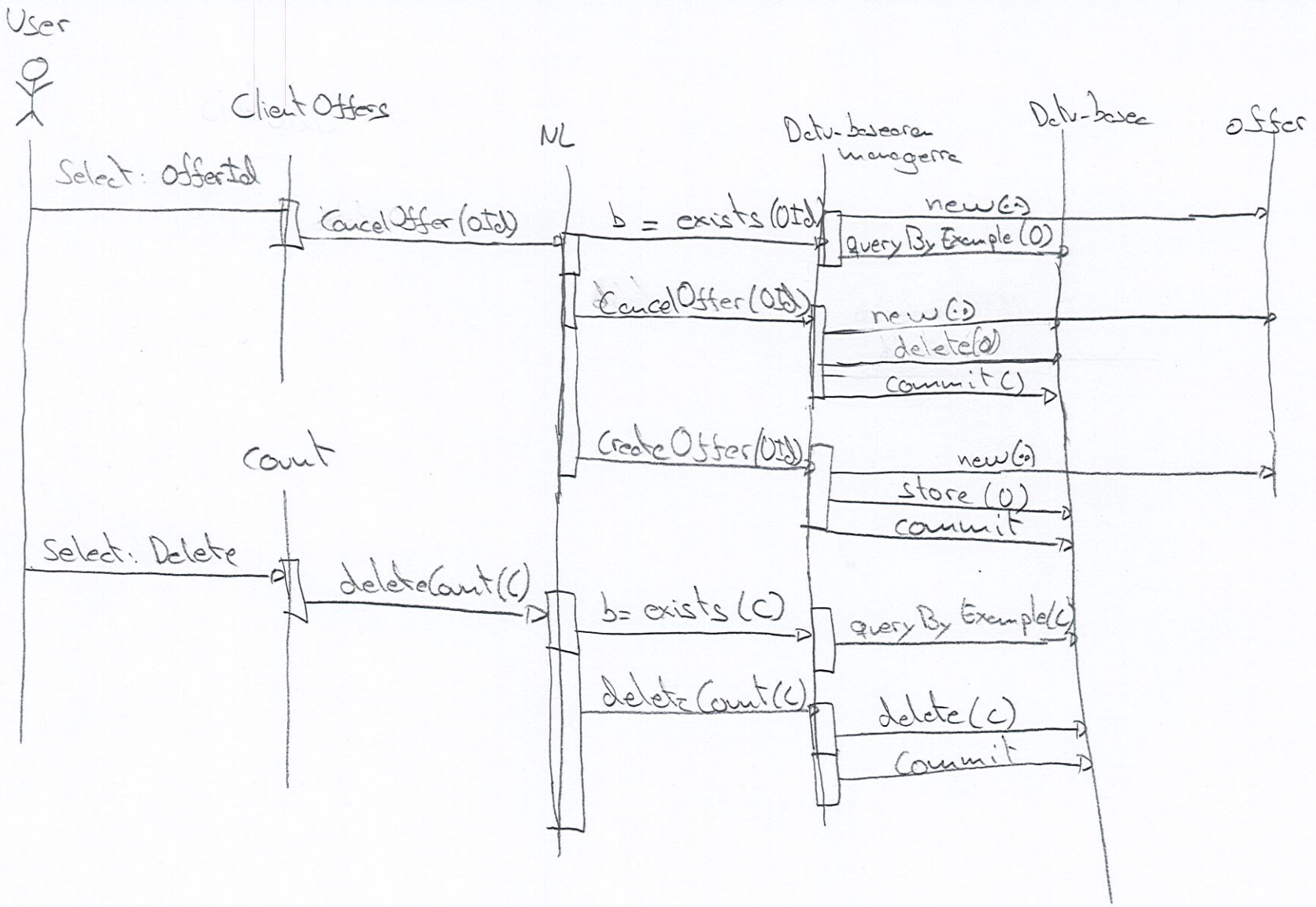
EBALUAZIO JARRAITUA			EBALUAZIO GLOBALA		
Jarduera	**Pisua**	**Minimoa**	**Jarduera**	**Pisua**	**Minimoa**
Aurkezpena (Filtro)					
3 marroi (Filtro)	15				
			PROEIKTUA DEFENTSAREKIN	50	50%
MARROIAK ETA AURKEZPENA	**15**	**2 marroi**			
2 azterketa (Puntuazioa)	10	30%			
AZTERKETA	**10**	**30%**			
Proiektuaren lehendabiziko iterazioa	15	30%			
Proiektuaren bigarrengo iterazioa	25	30%	AZTERKETA	50	30%
Proiektuaren defentsa eta galde-sorta	35	30%			
PROIEKTUAREN TOTALA	**75**	**50%**			
IRAKASGAIAREN TOTALA	100	50%	**IRAKASGAIAREN TOTALA**	100	50%


Owner
KandelkeSoftwGui
NL
Data-basea manager
Data-basea
RuralHouse
intro: RHId, Kokalaka, plazaKop, Telf, email
CreateRuralHouse (RH, RHId, KokAkut, plazaKop, Telf, email)
b = existsRA (RRId)
overlaps (RHId)
createRuralHouse (RH, RHId ...)
new(...)
store(RH)
commit()


USER
Login-Gui
NL
Datu-basea
Datu-basea (MySQL)
User
user; pass
getLogin(user, pass)
u = getUser(user, pass)
queryByExample(new User(null, ..., null))
new(user, pass)
Emaitza
Emaitza
list
Erregis-Gui
NL
Sartu:
erab Izena; Kontua; izena; abizena; [pass=Compass];
aukera
createUser(E,K,I,A,P,CP,Au)
b = existsUserName(erab Izena)
createUser(...)
new(...)
store(User)
commit()
check:
Pass=Compass


User
QueryAvailability
NL
Data_base_manager
Data_base
Offer
Valoration
RuralHouse
Select: offerId
bookOffer(OId)
b = exists(OId)
bookOffer(OId)
new(...)
QueryByExample(O)
new(..)
store(O)
commit
Intro: valoration
CreateValoration(val)
CreateValoration(val)
new(...)
store(val)
commit
Valoration
Select: favorite
addfavorite(RH)
b= exists(RH)
addfavorite(RH)
queryByExample(RH)
Store(RH)
commit


Owner
DeleteRuralHouseGui
NL
Data-Basaran manegerra
Data-basec
RuralHouse
intro: RuralHouseId
DeleteRuralHouse(RuralHouseId)
b= exists(RuralHouseId)
new(...)
queryByExample(RH)
DeleteRuralHouse(RHId)
delete(RHId)
commit()
Valoration
Admin
ValorationGui
new(...)
b= exists(vId)
queryByExample(V)
select: valorationId
deleteValoration(vId)
deleteValoration(vId)
delete(vId)
commit


User
Client Offers
NL
Data-bearer manager
Data-base
offer
Select: offerId
CancelOffer(oId)
b = exists(oId)
new(..)
queryByExample(o)
CancelOffer(oId)
new(..)
delete(o)
commit()
count
createOffer(oId)
new(..)
store(o)
commit
Select: Delete
deleteCount()
b= exists(c)
queryByExample(c)
deleteCount(c)
delete(c)
commit