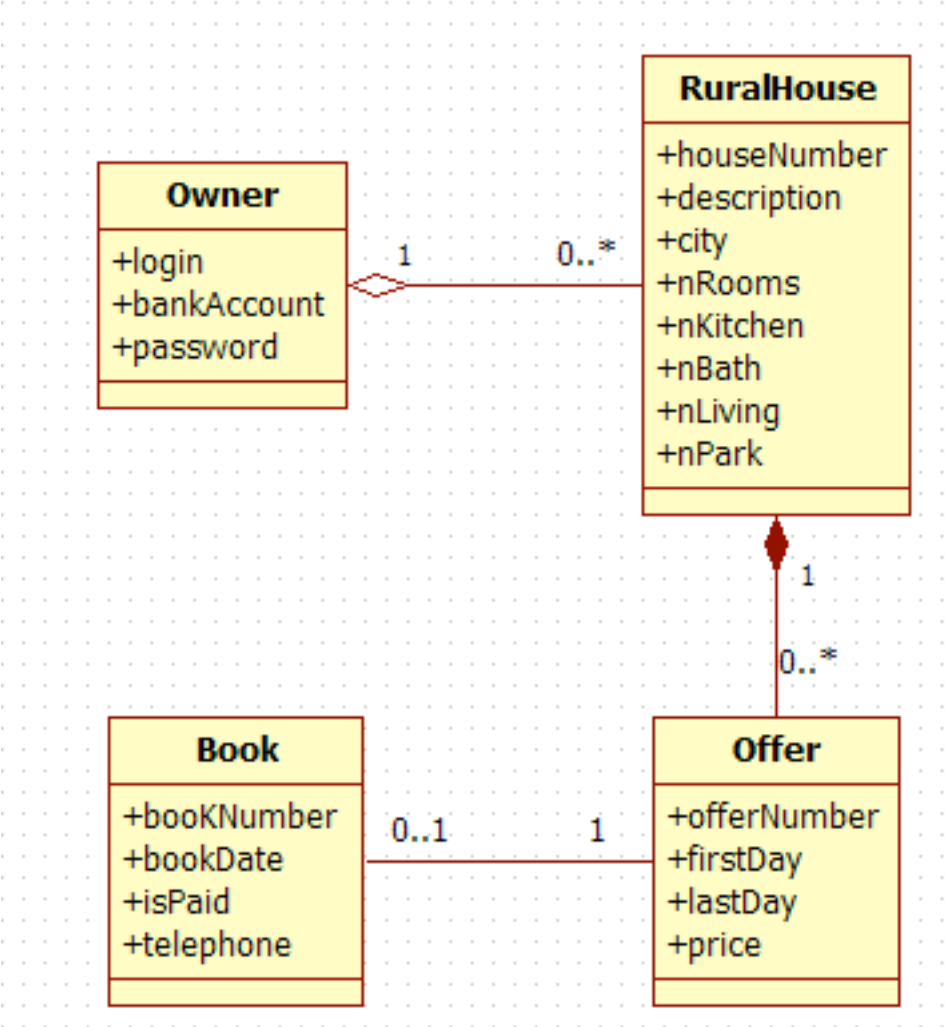
Dokumentu honetan landetxe aplikazioaren ebazpena aurkezten da, hurrengo atalak jarraituz.

1. Eskakizunaren Bilketa.
    a. Domeinuaren eredua.
    b. Erabilpen kasuen eredua.
    c. Gertaera fluxuak.
2. Diseinua.
    a. 3 erabilpen kasuen Sekuentzia diagramak.
3. Inplementazioa.
    a. Aplikazioaren egitura.
    b. Aplikazioa martxan jartzeko argibideak.

1. Eskakizunaren bilketa.

a. Domeinuaren eredua.



b. Erabilpen kasuen eredua.

c. Gertaera fluxuak

ERABILPEN KASUA: Identify
1. Jabeak izena eta pasahitza sartzen du
2. Sistemak izen eta pasahitza horrekin jaberen bat existen bada, sisteman sartzen uzten dio

Gertaeren fluxu alternatiboa:
1.- Ez dago jaberik izen horrekin. Amaiera.
2.- Emandako pasahitza ez dator bat jabearen izenarekinAmaiera.

ERABILPEN KASUA: Registry
1. Jabeak izena,pasahitza eta kontu zenbakia sartzen du
2. Sistemak Jabe bat sortzen du aurreko datuekin
3. Sistemak kontu bat sortu dela adierazten dio Jabeari

ERABILPEN KASUA: Add Rural House
1. Jabeak, herria, logela kopurua, bainuak, sukaldeak, jangelak, aparkaleku kopurua eta, nahi badu, etxearen deskribapena sartzen ditu.
2. Sistemak Landetxe berri bat sortzen du Jabe horrentzako eta jabeari aurkezten dio.

Gertaeren fluxu alternatiboa:
1.- Jabeak emandako sukalde kopurua bat baino txikiagoa izatea. Amaiera.
2.- Jabeak emandako logela kopurua hiru baino txikiagoa izatea. Amaiera.
3.- Jabeak emandako bainu kopurua bi baino txikiagoa izatea. Amaiera.

ERABILPEN KASUA: Remove Rural House
1. Sistemak Jabe horren landetxe guztiak aurkezten dizkio
2. Jabeak landetxe bat aukeratzen du.
3. Sistemak landetxe horretan zeuden Eskaera eta Erreserba guztiak ezabatzen ditu.Eskaera batentzako Erreserbaen bat eginda balego, Jabeari komunitzen zaio.
4. Sistemak Landetxea ezabatzen du.

ERABILPEN KASUA: Book paid
1. Sistemak Jabe horren landetxe guztiak aurkezten dizkio.
2. Jabeak etxe bat aukeratzen du.
3. Sistemak landetxe horretan dauden ordaindu gabeko erreserbak aurkezten dizkio Jabeari.
4, Jabeak erreserba bat aukeratzen du.
5. Sistemak erreserba hori egoera="ordainduta" jartzen du.

ERABILPEN KASUA: Create an Offer
1. Sistemak Jabe horren landetxe guztiak aurkezten dizkio.
2. Jabeak etxe bat aukeratzen du.
3. Jabeak eskaintzaren hasiera data eta egun kopurua sartuko du.
4. Sistemak Eskaintza berri bat sortuko du etxe horrentzako Jabeak sartutako datuekin.

ERABILPEN KASUA: Search Offers
1. Bezeroak herriaren izena ematen du.
2. Sistemak herri horretako landetxeak bilatu eta zerrenda bat itzultzen du.
3. Bezeroak landetxea aukeratzen du.
4. Bezeroak sarrera-data eta egun kopurua ematen ditu.
5. Sistemak eskaerak betetzen dituzten eskaintzak itzultzen ditu eta erreserba egiteko aukera ematen dio. Eskaintzarik ez balego, bezeroari jakinarazten zaio, eta egun hurbiletako eskaintzak aurkezten zaizkio.

Gertaeren fluxu alternatiboa:
1.- Ez dago landetxerik emandako kodearekin. Amaiera.
2.- Emandako herrian ez dago landetxerik. Amaiera.

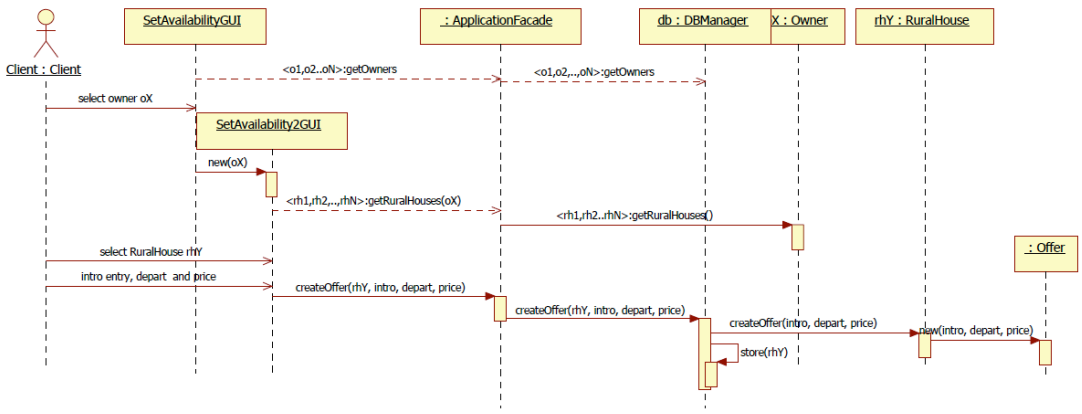
ERABILPEN KASUA: Realize a Book
1. Bezeroak etxearen kodea, sarrerako eguna eta alokatu nahi duen egun kopurua sartzen du.
2. Sistemak eskatu dituen egunetarako eskaintza librerik dagoen egiaztatzen du.
3. Eskaera Libre badago,Sistemak Erreserba bat sortuko du Eskaera horrentzako.Jarraian landetxearen jabearen kontu korrontea agertuko da, eta bezeroari telefono zenbakia eskatuko zaio.
4. Bezeroak telefonoa sartuko du eta ondoren erreserba egina geldituko da.
5. Eskaera Librerik ez badago,Sistemak Bezeroari epe horretarako eskaintza librerik ez dagoela jakinaraziko dio

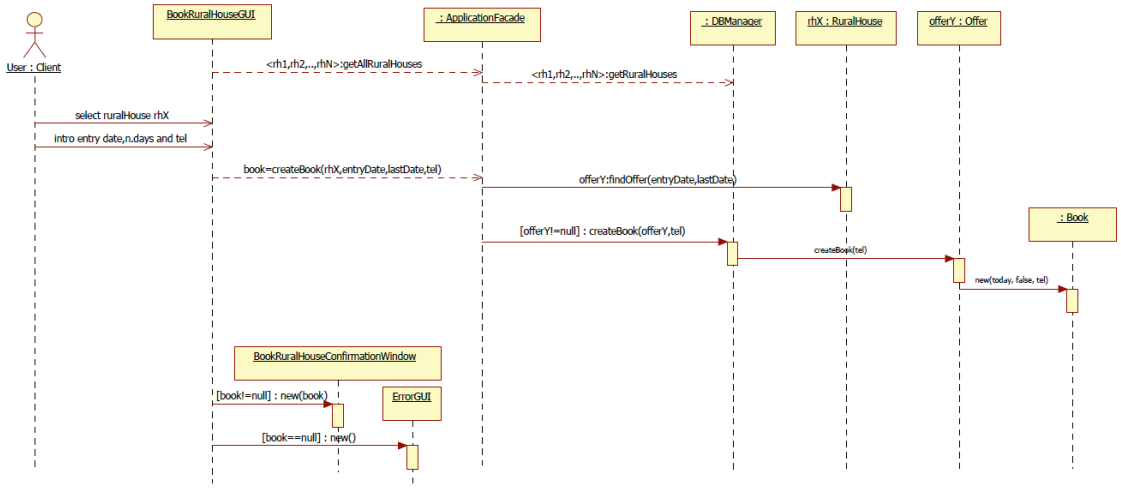
Gertaeren fluxu alternatiboa:
1.- Ez dago landetxearik kode horrekin. Amaiera.
2.- Ez dago emandako datekin bat egiten duen eskaintza librerik. Amaiera.

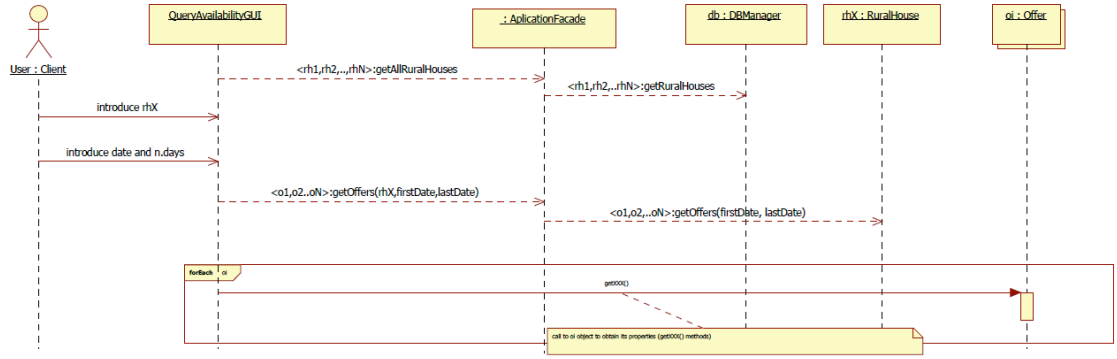
2. Diseinua.

a. 3 erabilpen kasuen Sekuentzia diagramak

Aplikazioa hiru mailako arkitektura batean diseinatuta dago. AWT/SWING lengoaia erabili da erabiltzaileren interfaze grafikoak definitzeko, negozio logikaren atzipena RMI-ren bitartez definitu da, eta persistentzia mailarako, objektuetan oinarritutako db4o datu basea erabili da. Negozio Logikaren eragiketa guztiak biltzeko, eta GRASP patroiaren gomendioak jarraituz, Kontrolatzaile klase bat definitu da eragiketa guztiekin. Jarraian sistemaren 3 erabilpen kasuen diagramak aurkezten dira;







3. Inplementazioa.

a. Aplikazioaren egitura.

Aplikazioa 6 paketetan antolatuta dago:

gui: Interfaze grafikoko klaseak
bussinessLogic: Negozio logikako klaseak
domain: Domeinu ereduan definitu diren klaseak

dataAccess: Pertsistentziako klaseak
configuration: konfiguraziozko parametroak gordetzen den paketea
exception: Salbuespenak dauden paketea

b. Aplikazioa martxan jartzeko argibideak.

Fitxategi honetan (Proiektua.zip) aplikazioaren inplementazioa eta aplikazioa martxan jartzeko behar dituzuen paketeak daude.

Exekutatzeko hurrengo pausoak jarraitu:

1. Java proiektu bat sortu (File->New->Java Project) eta “Create project from existing source” aukeratu. Pantaila honetan “ruralhouse” deskonprimitu duzuen karpetaren helbidea ipini beharko duzue.
2. Behar diren liburutegiak kargatu. Proiektuaren gainean ipini, eta eskuineko botoarekin Properties aukeratu. Agertzen den pantailan ezkerreko menuan “Java Build Path” aukeratu, eta eskuineko pantailan Libraries aukeran, “Add external Jar” sakatu. Kargatu behar diren liburutegiak, emandako fitxategian aurkeztu ditzakezue:

   a) JCalendar liburutegia, AdditionalLibraries karpetan.
   b) Db4o liburutegiaren klase guztiak, ObjectManager-7.14/lib dauden jar fitxategi guztiak.
3. Aplikazioa konfiguratu. configuration paketean dagoen Config fitxategian hurrengo aldagaiak konfiguratu:
   1. javaPolicyPath aldagaia, java.policy fitxategia dagoen helbidearekin(Proiektua.zip-ean ematen da).
   2. db4oFilename aldagaian, datu-basea gordeko den karpeta eta fitxategiaren izena. Derrigorrezkoa da ipinitako karpeta fitxategi sisteman existitzea.
   3. dataBaseOpenMode aldagaian, datu basea ireki nahi dugun moduan definituko dugu. Bi aukera daude:(1) *initialize*, datu basea hasieratzen da defektuzko balioekin, eta (2) *open*, datu basea irekitzen da dauden balioekin. Modu honetan, datu basea existitzen ez bada, berri bat sortzen da.

4. RMI zerbitzari urruna jaurti. "Run" businessLogic.RemoteServer klasea.
5. Interfazea jaurti. Run gui.StartWindow klasea. Klase honetan "*isLocal*" negozio logikaren atzipen aldagai bat konfiguratu daiteke. "*isLocal*" true konfiguratzen bada, negozio logika eta interfazea makina berdinean daudela esan nahi du, eta klase arrunta bat bezala atzituko da. Ordea, "*isLocal*" false konfiguratzen bada, orduan negozio logika eta interfazea makina desberdinetan daudela adierazten du, eta RMI-ren bitartez atzituko da.

OHAR: Inplementazioan, datak sartzeko Kai Toedlter-en JCalendar klasea erabiliko da. Klase honen erabilpena ezagutzea proiektuaren helburu bat da.