

PRÀCTICA Arquitectura del Software

1er Lliurament

QT12-13

- La solució de la pràctica s'ha d'entregar en un fitxer .pdf al Racó.
- Tots els diagrames s'han de poder llegir correctament.
- La data de lliurament de la pràctica és el 13/11/2012

Enunciat: Traveling

L'agència de viatges *Traveling* organitza viatges a mida des de Barcelona a ciutats de tot el món. Actualment, l'agència no disposa de cap sistema software i ens ha demanat que dissenyem una part d'un sistema software encarregat de gestionar els viatges que fan els seus clients.

L'agència vol enregistrar els seus clients i les ciutats on ofereix viatges. Dels clients sabem el seu dni (que els identifica), el nom, el telèfon i el nombre de viatges contractats. De les ciutats, sabem el seu nom (que les identifica), el nom de l'aeroport principal i el preu del vol a aquella ciutat amb origen a Barcelona. Cada ciutat disposa d'un conjunt d'hotels. Dels hotels sabem el seu nom, la descripció i el preu que cobren per una habitació (totes les habitacions d'un hotel tenen el mateix preu). Els hotels s'identifiquen pel nom de l'hotel i pel nom de la ciutat on estan localitzats. Hi ha tres tipus diferents d'hotels, els de categoria standard, els de categoria low cost i els de categoria superior. Els hotels low cost tenen un descompte en el preu de l'habitació mentre que els de categoria superior tenen un recàrrec. Els hotels disposen d'un conjunt d'habitacions que poden oferir als seus clients. Les habitacions s'identifiquen pel seu número dins de l'hotel. Els clients poden contractar viatges a ciutats per un període de temps. Evidentment, no poden existir dos viatges pel mateix client en períodes de temps solapats. A més, opcionalment, els clients poden fer la reserva d'una habitació a la ciutat destí pel període del viatge.

De les moltes funcionalitats que oferirà el sistema, ens han demanat dissenyar tres casos d'ús. A continuació disposeu d'una descripció textual de les funcionalitats, del diagrama de casos d'ús, de la descripció textual dels casos d'ús i de l'esquema conceptual de l'especificació:

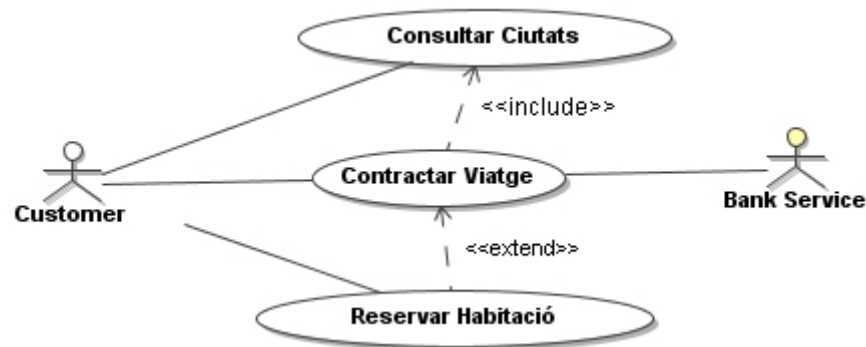
Descripció de les funcionalitats:

- Contractar viatge: el personal de l'agència (l'agent) fa la petició de la contractació d'un viatge. El sistema li retornarà una llista amb el nom de totes les ciutats on es poden fer viatges i el preu del vol a aquella ciutat. L'agent

selecciona una ciutat, introdueix el client i el període del viatge. El sistema, si no hi ha cap error, enregistrarà el viatge. L'agent indicarà si vol reservar una habitació pel viatge. Si és així, el sistema mostrarà el nom i preu dels diferents hotels que tenen habitacions lliures en aquell període. L'agent indicarà l'hotel en el que vol fer la reserva i el sistema reservarà una habitació qualsevol de l'hotel per a aquell viatge. L'agent indicarà les dades del pagament (mitjançant targeta de crèdit) i el sistema, si no hi ha cap error, cobrarà l'import total del viatge (preu del vol més opcionalment preu de l'habitació).

- Consultar ciutats: el sistema retornarà una llista amb el nom de totes les ciutats on es poden fer viatges i el preu del vol a aquella ciutat.
- Reservar habitació: l'agent introduirà el client i la data inici del viatge. El sistema mostrarà el nom i preu dels diferents hotels que tenen habitacions lliures en aquell període. L'agent indicarà l'hotel en el que vol fer la reserva i el sistema, si no hi ha cap error, reservarà una habitació qualsevol de l'hotel per a aquell viatge.

Diagrama de casos d'ús:



Descripció textual dels casos d'ús:

Nom: Contractar viatge

Resum de la funcionalitat: permet contractar un viatge i, opcionalment, reservar una habitació.

Flux d'esdeveniments principal:

- 1) El client vol contractar un viatge i li comunica a l'agent.
- 2) L'agent indica al sistema que vol iniciar la contractació d'un viatge.
- 3) S'executarà el cas d'ús Consultar Ciutats que retornarà una llista amb el nom de totes les ciutats on es poden fer viatges i el preu del vol a aquella ciutat.
- 4) L'agent selecciona una ciutat, introdueix el dni del client i el període del viatge.
- 5) El sistema enregistrarà el viatge.
- 6) Opcionalment, l'agent indica al sistema que vol fer la reserva d'una habitació pel viatge.
- 7) Opcionalment, s'executarà el cas d'ús Reservar Habitació (amb el dni del client i la data d'inici del viatge).

- 8) El sistema mostra el total a pagar (preu del vol més, opcionalment, el preu de l'habitació en el període reservat) i demana les dades del pagament.
- 9) L'agent introdueix les dades del pagament (dni, número de targeta i data de caducitat) i accepta fer el pagament.
- 10) El sistema invoca a un servei de l'entitat bancària per cobrar el viatge. Si el pagament s'ha autoritzat, s'informa a l'usuari.

Flux d'esdeveniments alternatiu:

- 3) El sistema mostra un error si no hi ha ciutats al sistema.
- 5) El sistema mostra un error si:
 - El client no existeix
 - El període de temps no és vàlid.
 - El client ja té un viatge per a un període solapat amb les dates introduïdes.
- 7) El sistema mostra un error si no hi ha cap hotel disponible.
- 10) El sistema mostra un error si el servei no està disponible o si el pagament no s'ha autoritzat, informa a l'agent i s'acaba el cas d'ús.
- 1-4) L'usuari cancel·la la contractació del viatge. El cas d'ús acaba.
- 5-7) L'usuari cancel·la la reserva de l'habitació i el sistema continua amb el pas 8)

Nom: Consultar ciutats

Resum de la funcionalitat: permet consultar totes les ciutats que hi ha enregistrades al sistema

Flux d'esdeveniments principal:

- 1) L'agent indica al sistema que vol consultar les ciutats.
- 2) El sistema retornarà una llista amb el nom de totes les ciutats on es poden fer viatges i el preu del vol a aquella ciutat.

Flux d'esdeveniments alternatiu:

- 2) El sistema mostra un error si no hi ha ciutats al sistema.

Nom: Reservar habitació

Resum de la funcionalitat: permet reservar una habitació per a un client, ciutat i període de temps.

Flux d'esdeveniments principal:

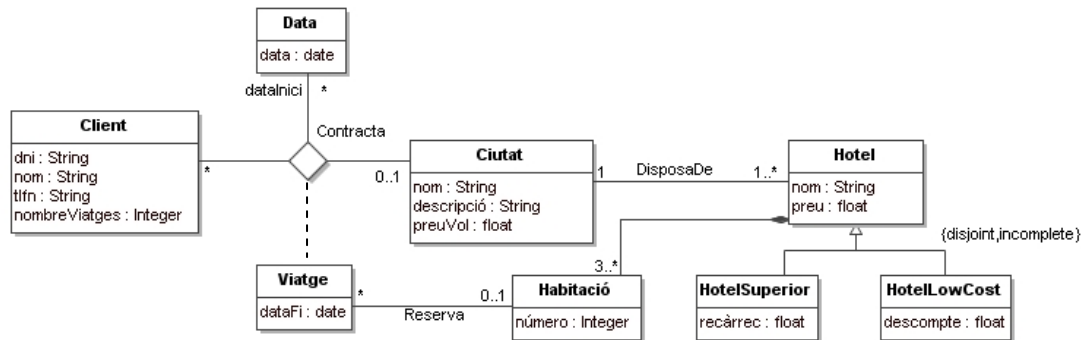
- 1) L'agent indica al sistema que vol reservar una habitació.
- 2) L'agent introdueix el dni del client i la data inici del viatge.
- 3) El sistema mostrarà la ciutat del viatge, la data fi del viatge, el nom i preu dels diferents hotels que tenen habitacions lliures en aquell període i ciutat.
- 4) L'agent indicarà l'hotel en el que vol fer la reserva.
- 5) El sistema reservarà una habitació qualsevol de l'hotel per a aquell viatge.

Flux d'esdeveniments alternatiu:

- 3) El sistema mostra un error si:
 - No hi ha cap hotel disponible.
 - No hi ha un viatge pel client i data inici.
 - Ja hi ha una habitació reservada pel viatge del client.

Nota: Per simplificar suposarem que quan la reserva d'una habitació es fa fora del cas d'ús de contractació del viatge, el pagament es farà en efectiu.

Esquema conceptual de l'especificació:



Restriccions d'integritat textuals:

1. Client s'identifica per dni.
2. Data s'identifica per date.
3. Ciutat s'identifica per nom.
4. Hotel s'identifica per nom (de Ciutat) i nom.
5. Habitació s'identifica per nom (d'Hotel) i número.
6. La data d'inici d'un viatge ha de ser menor que la data de fi del viatge.
7. No poden existir dos viatges pel mateix client solapats en el temps.
8. Una habitació no pot estar reservada per dos viatges en períodes solapats.
9. L'habitació reservada d'un viatge ha de ser d'un hotel de la mateixa ciutat on es fa el viatge.
10. Els atributs preuVol, preu, recàrrec, descompte i número han de ser més grans que 0.
11. L'atribut nombreViatges de Client serà igual al nombre de viatges contractats pel client.

Altres restriccions no rellevants pel nostre disseny....

Després de fer l'assignació de responsabilitats a capes pel cas d'ús *Contractar viatge*, hem obtingut els següents contractes de la capa de domini:

Contractes Capa Domini:

context CapaDomini :: obtéCiutats(): Set(TupleType(nom: String, preuVol: float))

exc *noHiHaCiutats*: no hi ha ciutats enregistrades al sistema.

post *result*= obté el nom i el preu del vol de totes les ciutats existents al sistema.

context CapaDomini :: enregistraViatge(dni: String, dataIni: date, dataFi: date, nom: String)

pre *ciutatExisteix*: La ciutat amb nom *nom* existeix.

pre *datesCorrectes*: Les dates són correctes i dataIni és menor que dataFi.

exc *clientNoExisteix*: el client amb dni *dni* no existeix.

exc *jaTéViatge*: el client ja té un viatge contractat per a un període solapat amb les dates introduïdes.

post creacióViatge: es crea una instància de viatge pel client ciutat i dates.
post nouViatge: s'incrementa el nombre de viatges contractats pel client.
post emmagatzemaDades: s'emmagatzema a la capa de domini el dni del client, la data d'inici, la data de fi, el nom de la ciutat i el preu del vol a la ciutat.

context CapaDomini :: mostraHotelsLliures():Set(TupleType(nom: String, preuHb: float))
exc hotelsNoLliures: no hi ha cap hotel disponible pel viatge.
post result= obté el nom i el preu de l'habitació dels hotels que tenen habitacions lliures en la ciutat i període del viatge. El preu d'una habitació és igual a: 1) el preu de l'hotel pel cas dels hotels standards, 2) el preu de l'hotel més recàrrec pels hotels superiors i 3) el preu de l'hotel menys el descompte pels hotel low cost. Aquest preu s'ha de multiplicar pel nombre de dies del període.

context CapaDomini :: reservaHabitació (nomH: String) : float
pre hotelExisteix: L'hotel amb nom *nomH* existeix i està lliure a les dates del viatge.
post creacióReserva: es crea una instància de l'associació *Reserva* entre el viatge i una habitació disponible qualsevol del hotel en el període del viatge.
post emmagatzemaDades: s'emmagatzema el preu total de l'habitació reservada.
post result= preu del vol + preu total de l'habitació.

context CapaDomini :: pagament (numTarg:String, dataCad: date): boolean
exc serveiNoDisponible: el servei no està disponible.
post pagament: el sistema crida a l'operació *autoritza(dni, numTarg, import, dataCad, dAvui):Boolean* del servei Bank Service amb les dades de la targeta i el preu total del viatge.
post result= cert si s'ha pogut fer el pagament o fals en cas contrari.

Es demana:

1.- Hibernate és una eina de lliure distribució que soluciona el problema de la coexistència entre els dos models de dades existents en un sistema software (l'utilitzat pel programa (orientació a objectes)) i l'utilitzat per la base de dades (model relacional). Es demana que utilitzeu Hibernate (amb anotacions) per definir una part de l'esquema conceptual (les classes *Hotel* i *Habitació* amb els seus atributs, l'associació entre elles i els getters i els setters de cada classe) i la part corresponent de l'esquema de la BD. En concret, es demana el codi en Java de les classes, l'esquema de la BD i un programa de prova que utilitzi les operacions definides a les classes. Podeu trobar molta informació sobre Hibernate a internet. Alguns links que us poden ser d'utilitat són: <http://www.davidmarco.es/tutoriales/hibernate-reference/> i <http://www.youtube.com/watch?v=GINvxAaXDbY&feature=related> (són 18 videos on es descriu com instal·lar i configurar Hibernate i com connectar-lo amb Eclipse i la BD. A més podeu veure com utilitzar Hibernate amb anotacions). La instal·lació i configuració de Hibernate i com connectar-lo amb Eclipse/Netbeans i PostgreSQL també la podeu trobar en el manual que us adjunto.

2.- Diagrames de seqüència de les operacions de la capa de domini corresponents al cas d'ús Contractar viatge. Heu de considerar que aquest cas d'ús utilitza els casos d'ús

Consultar ciutats i Reservar Habitació i que es valorarà especialment la reutilització dels mateixos.

3.- Diagrama de classes de la capa de domini. Heu de fer servir un controlador de cas d'ús per guardar l'estat del cas d'ús.

4.- (Competència transversal) Justifiqueu de forma raonada els patrons de disseny que heu aplicat en aquesta pràctica.