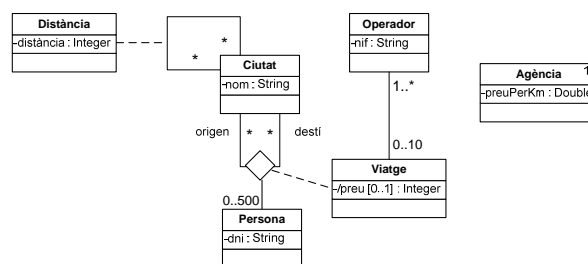


Unit 7. Presentation Layer Design

Exercise 4 Solution

1

Presentation Layer Design – Exercise 4



R. Textuals:

Claus: (*Ciutat*, *nom*); (*Persona*, *dni*); (*Operador*, *nif*)

RT1. Les ciutats *origen* i *destí* d'un viatge són diferents

RT2. Una persona no pot tenir més de 100 viatges

...altres restriccions que no són d'interès per al problema

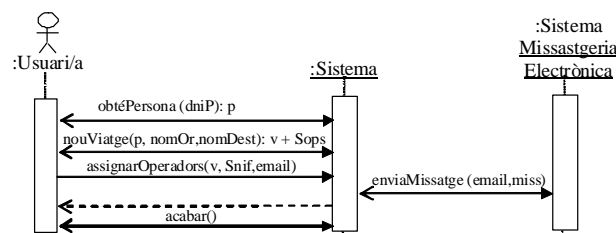
Informació derivada:

ID1. El *preu* d'un viatge és igual a la *distància* establerta entre la seva ciutat origen i destí, multiplicada pel *preuPerKm* estipulat per l'agència. Si les dues ciutats no estan directament relacionades per *Distància*, l'atribut *preu* té valor nul.

2

Presentation Layer Design – Exercise 4

1. L'usuari introdueix el dni de la persona i prem <<OK>>.
2. A continuació, l'usuari introdueix les dues ciutats i prem <<OK>>.
3. El sistema enregistra el viatge i mostra en una llista tots els operadors amb menys de 10 viatges assignats que, per tant, poden ser assignats com a operadors potencials a aquest nou viatge.
4. Finalment, l'usuari marca en aquesta llista els operadors potencials d'aquest viatge que desitja, introdueix el mail del client i prem <<OK>>.
5. El sistema enregistra els operadors del viatge, informa a l'usuari i invoca al sistema extern de missatgeria electrònica per enviar un mail amb les dades del viatge al client.
6. L'usuari confirma la recepció del missatge i s'acaba el cas d'ús.
7. Si es produeix algun dels errors possibles, es mostra un missatge i acaba el cas d'ús.



3

Presentation Layer Design – Exercise 4

context CapaDomini::obtéPersona (dniP:String)

exc personaNoExisteix: la persona amb dni *dniP* existeix.

post self.persona = persona amb dni *dniP*.

context CapaDomini::nouViatge (nomOr: String, nomDest: String): Sops:Set(string)

pre ciutatsDiferents: les ciutats origen i destí són diferents. (RT1)

exc ciutatOrigenNoExisteix: la ciutat amb nom *nomOr* no existeix. (R. referencial)

exc ciutatDestíNoExisteix: la ciutat amb nom *nomDest* no existeix. (R. referencial)

exc viatgeExisteix: el viatge per la persona, ciutat origen i destí existeix. (R. clau viatge)

exc personaAmbMoltsViatges: la persona ja té 100 viatges. (RT2)

exc ciutatsAmbMoltsViatges: les ciutats origen i destí ja tenen 500 viatges. (R. gràfica)

post viatgeCreat: crea una instància de viatge de *p* entre *nomOr* i *nomDest* i s'assigna el preu.

post self.viatge = viatge creat.

post result = nif dels operadors que tenen assignats menys de 10 viatges.

context CapaDomini::assignarOperadors (Snif: Set(String), email:String)

pre operadorsExisteixen: els operadors de Snif existeixen. (R. referencial)

pre operadorsSenseMaxViatges: els operadors de Snif no tenen 10 viatges assignats. (R. gràfica)

pre operadorsNoRepetits: si hi ha operadors a Snif, no hi ha repetits. (R. implícita no repetits associacions)

pre viatgeAmbOperadors: hi ha operadors a Snif. (R. gràfica)

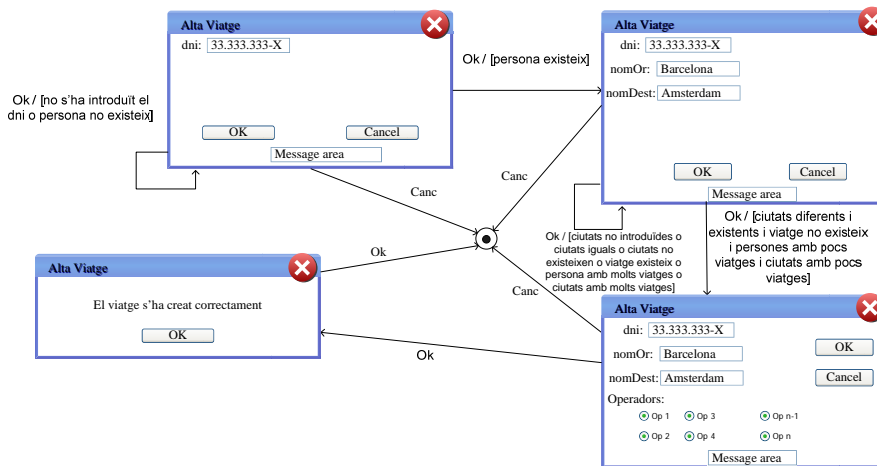
post operadorsAssociats: per tot nif *n* dins de *Snif*, associa el viatge *v* amb l'operador de nif *n*.

post enviaMail: s'invoca l'execució de l'operació *enviaMissatge* del sistema de *MissatgeriaElectrònica* amb l'email i el missatge de text amb les dades del viatge.

4

Presentation Layer Design– Exercise 4

(a) Definiu el model navegacional de cas d'ús (descripció d'alt nivell)

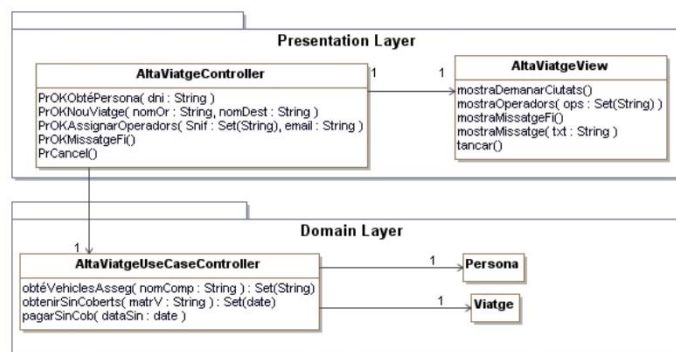


El botó ok estarà desactivat fins que es seleccioni un operador

5

Presentation Layer Design– Exercise 4

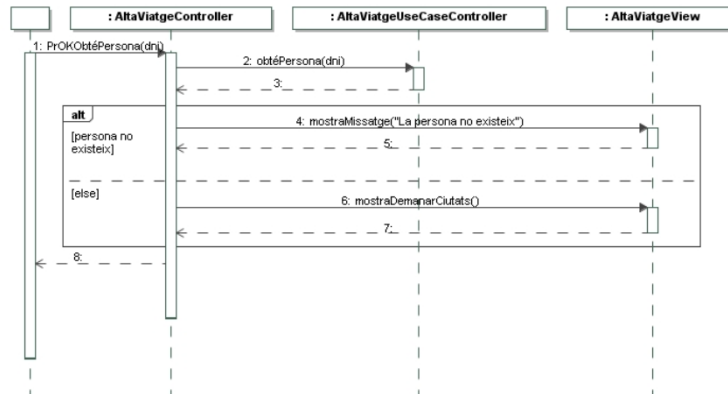
(b) Diagrama de classes de la Capa de Presentació i de Domini (en aquesta capa només cal que definiu els controladors)



6

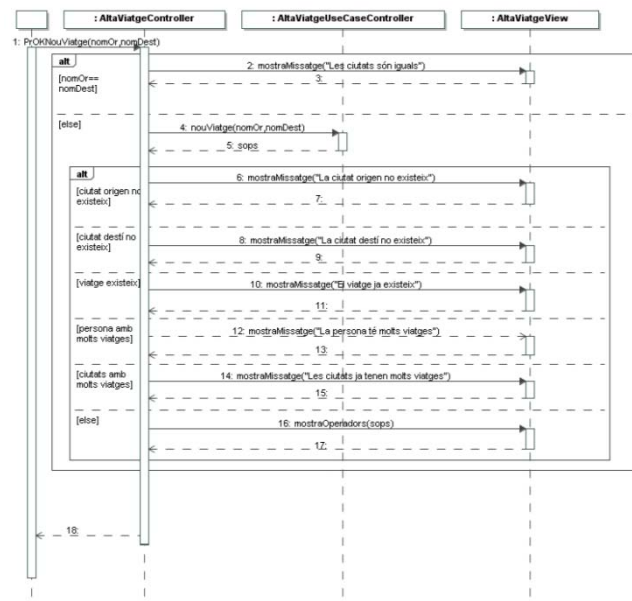
Presentation Layer Design– Exercise 4

(c) Diagrames de seqüència de les operacions dels controladors de la Capa de Presentació.



7

Presentation Layer Design– Exercise 4



8

Presentation Layer Design– Exercise 4

