Javier Delgado del Cerro
Javier López Cano

SECOND ASSIGMENT

Class diagram – Design choices

System

As the users should be able to be host and guest at the same time, we have created just one class called *RegisteredUser*, and the system has three lists which are the host, guests and both users. This way, the difference between those users will be set in the code, and the users can have any of the three roles.
Non registered users doesn't need a class because the limitations on the searches, for example, is set in the code.
The system has also a method called *cancelOffer* which is called when the it detects that more than five days have passed since the administrator asked for changes on an offer, and the user has not modified it, or when the start date of a holiday offer has passes and the offer wasn't reserved. This method calls *denyOffer*, which changes the offer status to -2 (we use this function instead of a setter because we may want to add an exaplanation of why the offer is discarded, or to notify the host, for example), and in case the offer is a holiday offer, it is removed from the system. If the offer is a living offer we could just call *denyOffer* method, but we prefer to use cancelOffer too because of consistency reasons.

House

Currently, the house is composed by the class *Characteristic* because it more visual to understand it, but on the actual implementaction we will use a HashMap to do that

RegisteredUser

The RegisteredUser class has a method *addOffer(o: Offer, s: Integer)* which allows us to add offers to his bought offers if he is a guest, to his on sale offers list, in case he is a host, or to any of them if he has both roles. We distinguish between the list by using an Integer.
The methods *seeHisotry* and *seeOffers* let us visualize the lists of bought and on sale offers previously mentioned.
The integer *status* shows whether the user is logged in, baned, or none of them, and it is changed my *changeStatus* method. For example, if the paymentSystem detects a fake credit card, he needs to ban the user, and he uses *changeStatus* for that.
In case the user is banned, we use *changeCreditCard* to change his credit card number, and also to change his status to a regular one (logged in or simply registered).

Comments

There are three kinds of comments.

- Text comments that can answer to another text comment.
- Rating: a mark from one to five.
- ChangeComment: a comment made by the administrator to let the host know what he should change if he wants his offer to be approved, or why was he his offer denied.

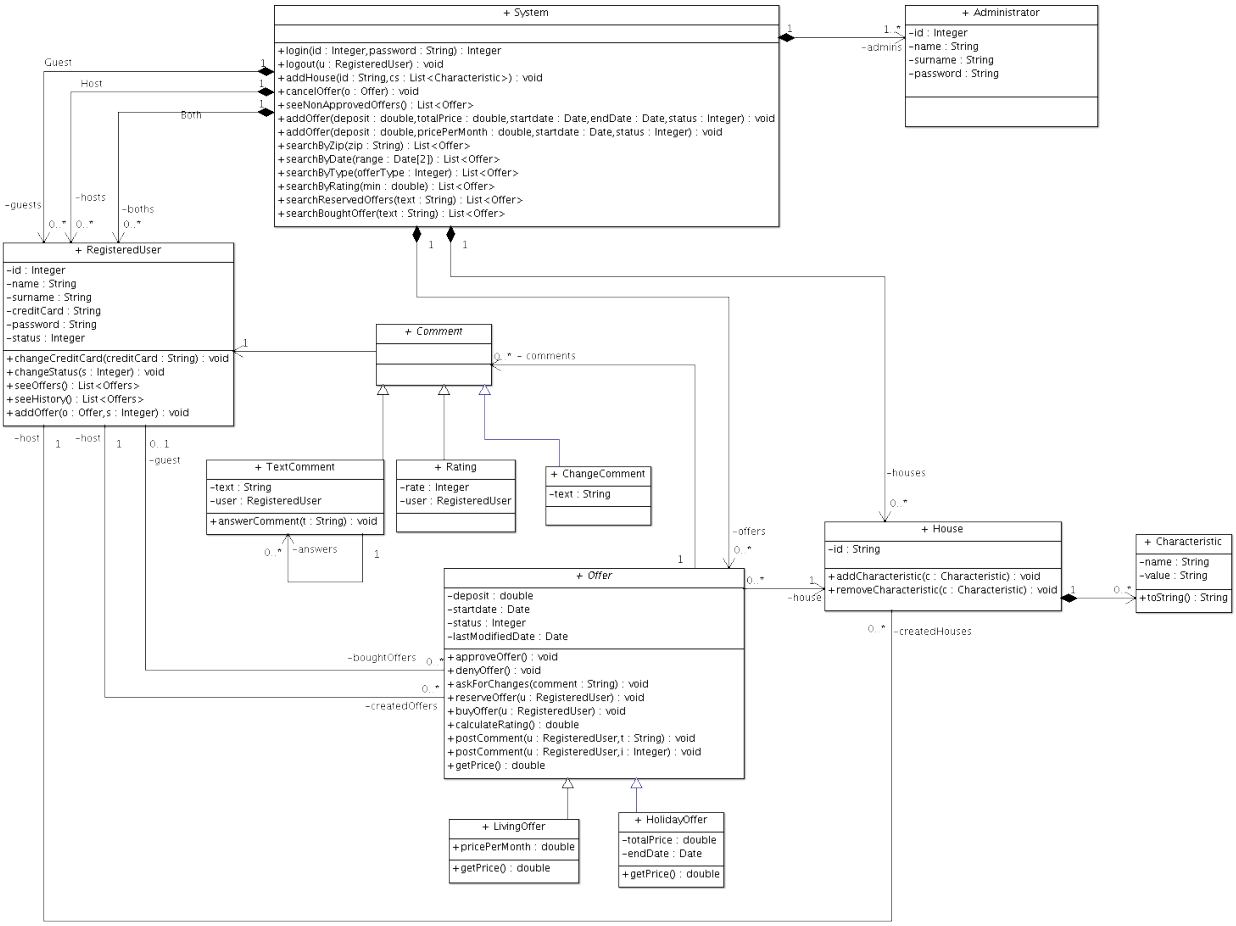
Offer

The integer *status* shows whether the offer is available, denied, reserved, bought, or if it needs some changes.
Once the administrator asks for changes in an offer, he needs to include some text, and a *ChangeComment* will be created to let the host know what he sould change. Those comments are not visible to any other user.

Javier Delgado del Cerro
Javier López Cano

We have included a *lastModifiedDate* on the offer in order to know when the offer was changed, denied, reserved (to detect that the user needs to pay), bought, or asked for changes (to detect that the user needs to modify the offer).

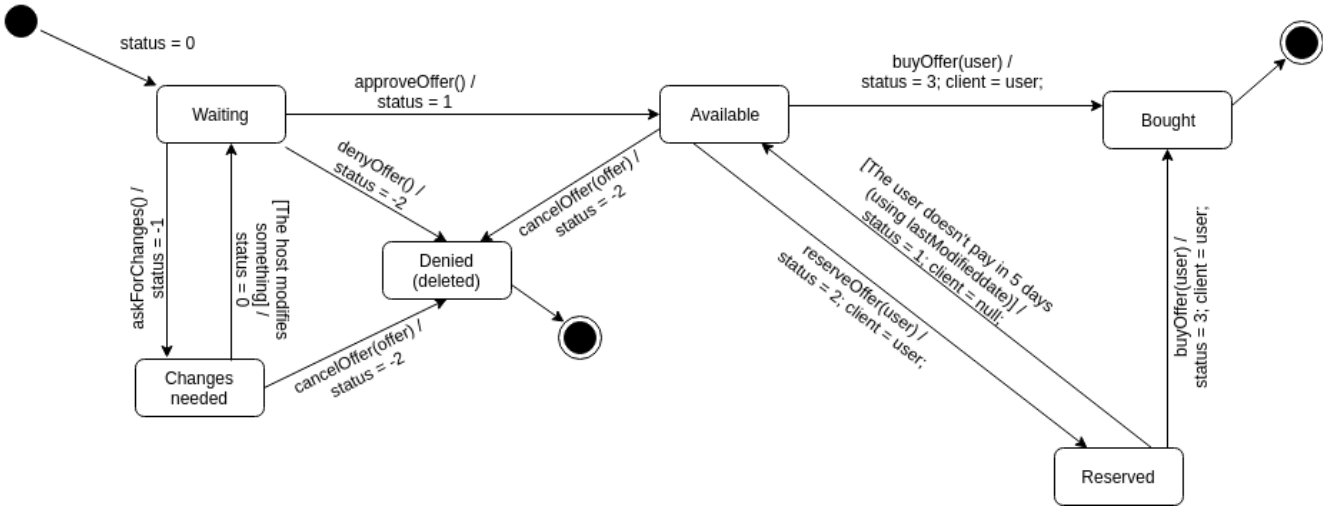
The method *calculateRating* calculates the average rating of the offer using the array of comments it has.



Javier Delgado del Cerro
Javier López Cano

State diagrams

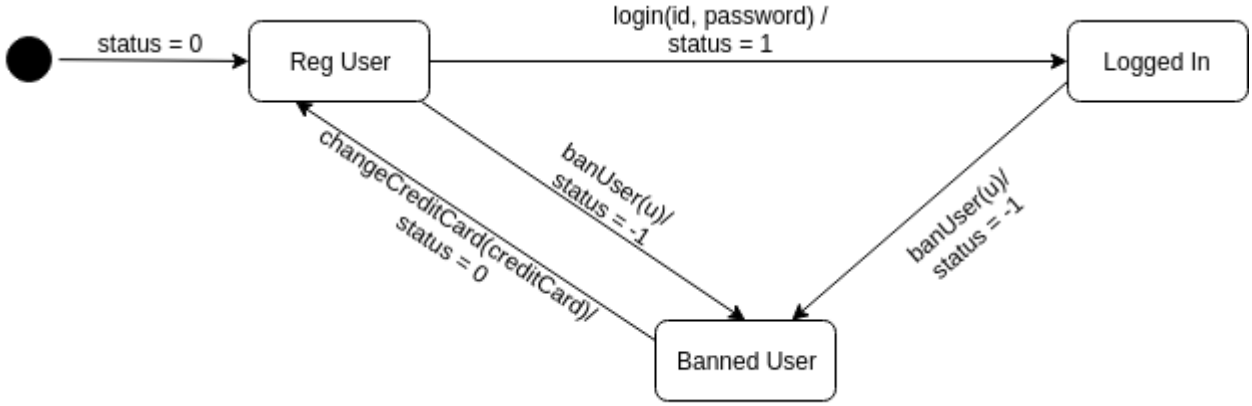
Offer state diagram



As we don't really know how this will be implemented in code, we suppose that cancelOffer is called automatically once five days have passed since the changes were asked or once the offer start date passes.

Registered user state diagram

As explained on the user class, *changeCreditCard* also modifies the user status.

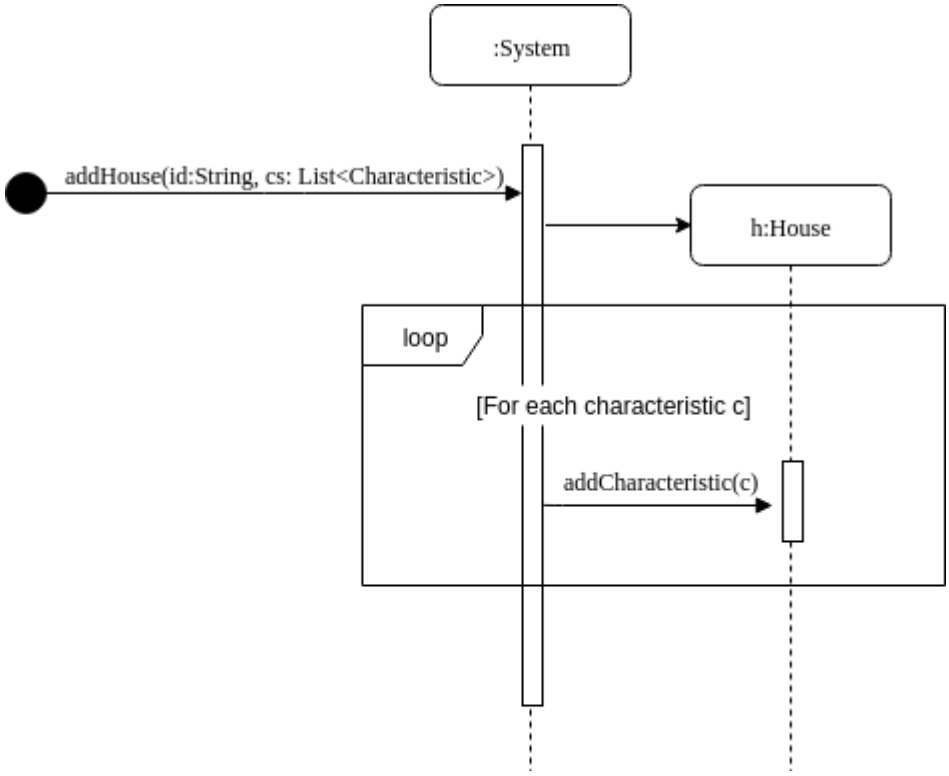


Javier Delgado del Cerro
Javier López Cano

Sequence diagrams

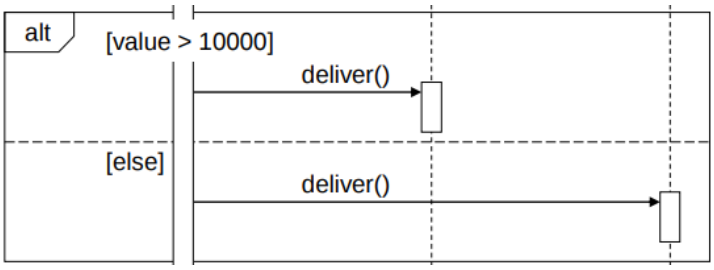
Add house secuence diagram

The function addHouse receives an id and a list of characteristics, and it creates a new house and insert each one of those characteristics to the created house.

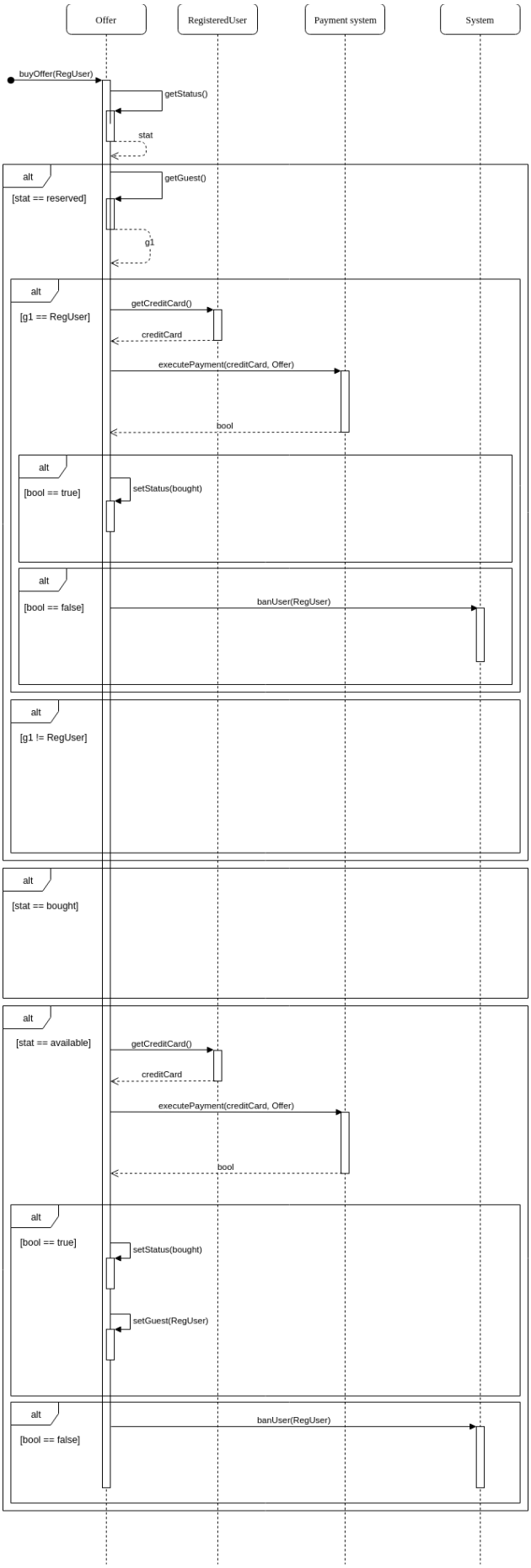


Buy offer sequence diagram

In the offer sequence diagram, the alternatives shold be displayed as an if and an else (as shown down), but as draw.io does not implement this structure, we have just used two alternatives (one for each condition) instead of one.



Javier Delgado del Cerro
Javier López Cano

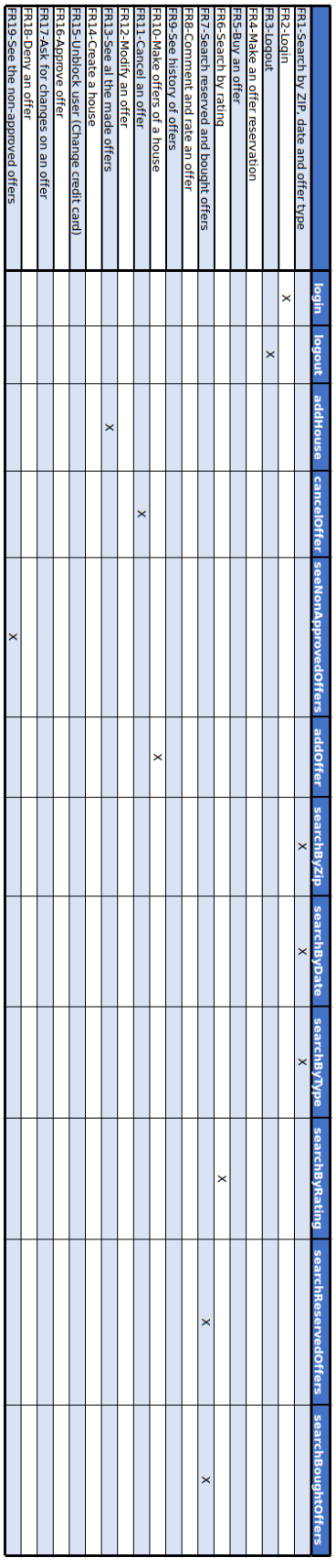


Javier Delgado del Cerro
Javier López Cano

Traceability matrix

In the traceability matrix, the modify offer functional requirement would be also affected by all the offer setters.

As it didn't fit in a sheet, we have divided it. This first one has all the methods from system class, and the sencond one includes the rest of the classes.



	login	logout	addHouse	cancelOffer	seeNonApprovedOffers	addOffer	searchByZip	searchByDate	searchByType	searchByRating	searchReservedOffers	searchBoughtOffers
FR1-Search by ZIP, date and offer type							X	X	X			
FR2-Login	X											
FR3-Logout		X										
FR4-Make an offer reservation												
FR5-Buy an offer												
FR6-Search by rating										X		
FR7-Search reserved and bought offers											X	X
FR8-Comment and rate an offer												
FR9-See history of offers												
FR10-Make offers of a house						X						
FR11-Cancel an offer				X								
FR12-Modify an offer												
FR13-See al the made offers			X									
FR14-Create a house												
FR15-Unblock user (Change credit card)												
FR16-Approve offer												
FR17-Ask for changes on an offer												
FR18-Deny an offer												
FR19-See the non-approved offers					X							

Javier Delgado del Cerro
Javier López Cano

	User methods				Offer methods						Comment methods		House methods	
	changeCreditCard	changeStatus	seeOffers	seeHistory	addOffer	approveOffer	denyOffer	askForChanges	reserveOffer	buyOffer	calculateRating	postComment	answerComment	addCharacteristic	removeCharacteristic
FR1-Search by ZIP, date and offer type															
FR2-Login		x													
FR3-Logout		x													
FR4-Make an offer reservation									x						
FR5-Buy an offer										x					
FR6-Search by rating											x				
FR7-Search reserved and bought offers															
FR8-Comment and rate an offer												x	x		
FR9-See history of offers				x											
FR10-Make offers of a house					x										
FR11-Cancel an offer															
FR12-Modify an offer															
FR13-See all the made offers			x												x
FR14-Create a house														x	
FR15-Unblock user (Change credit card)	x														
FR16-Approve offer		x				x									
FR17-Ask for changes on an offer								x						x	
FR18-Deny an offer							x								
FR19-See the non-approved offers															