Miguel Ángel Sánchez
Juan Velasco

miguelangel.sanchezm@estudiante.uam.es
juan.velascoi@estudiante.uam.es

Object Oriented Design

Application: **RentingJ&MA**

Date: 05/03/18

Miguel Ángel Sánchez   miguelangel.sanchezm@estudiante.uam.es
Juan Velasco   juan.velascoi@estudiante.uam.es

1. Class Diagram

Miguel Ángel Sánchez
Juan Velasco

miguelangel.sanchezm@estudiante.uam.es
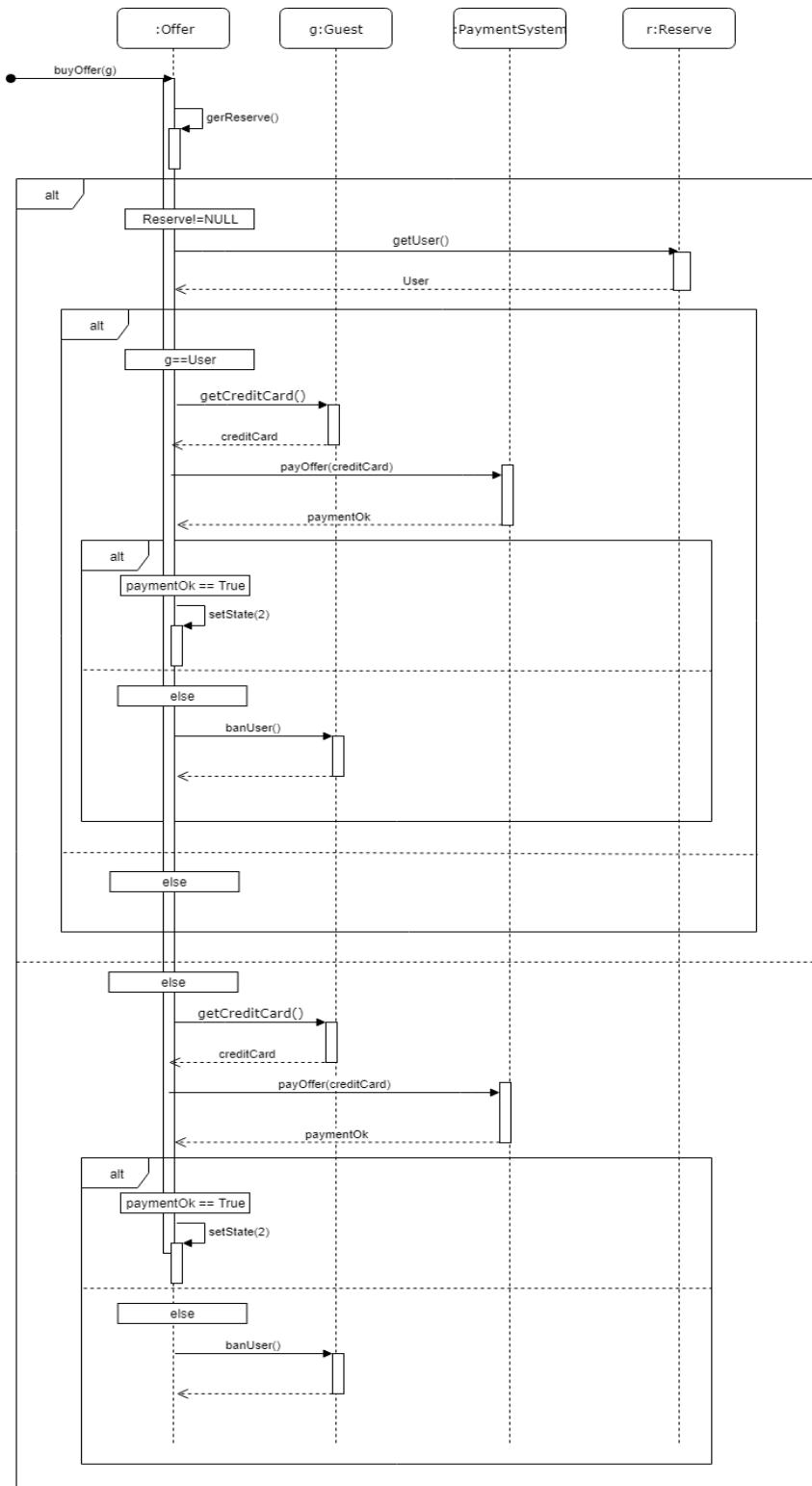*juan.velascoi@estudiante.uam.es*
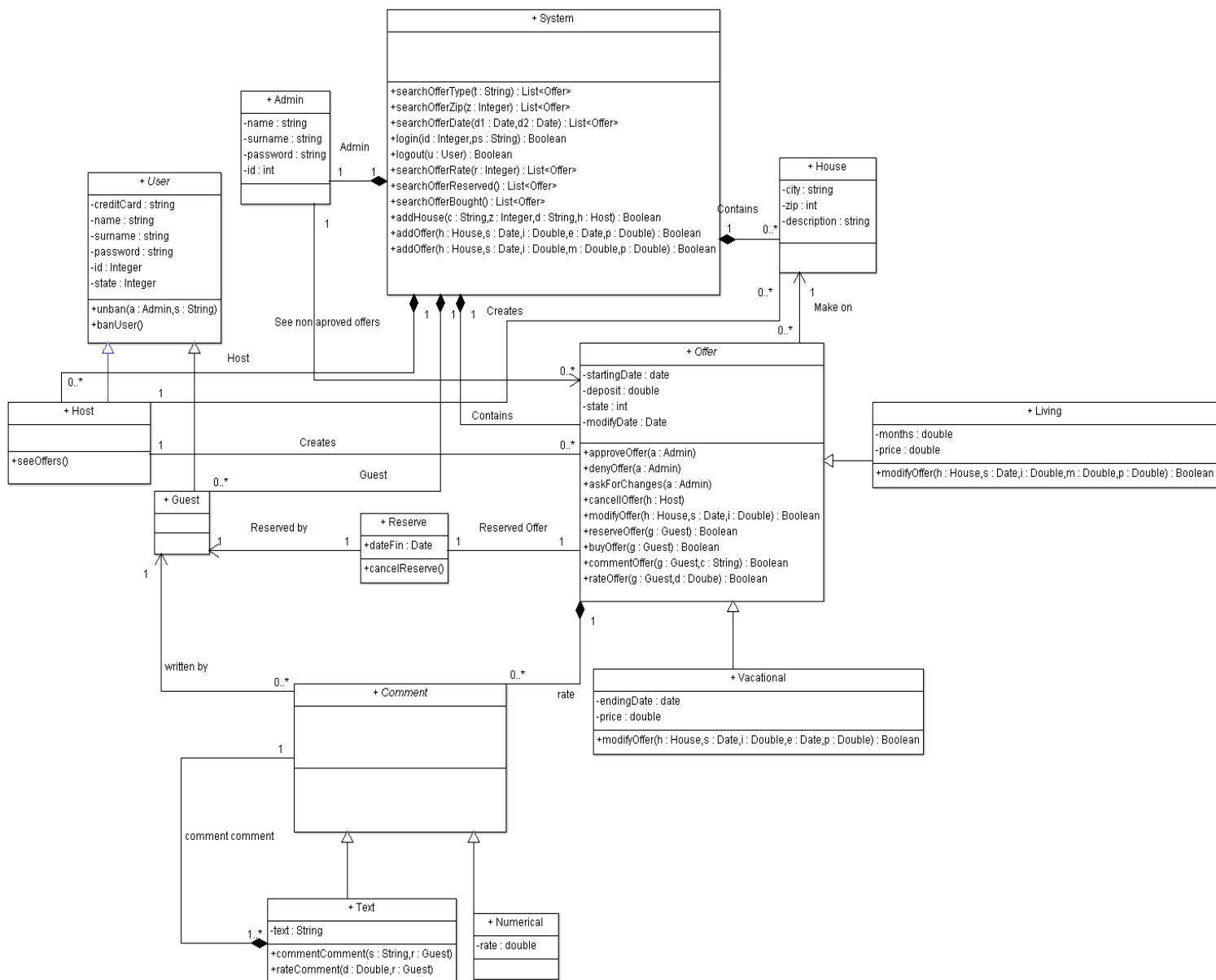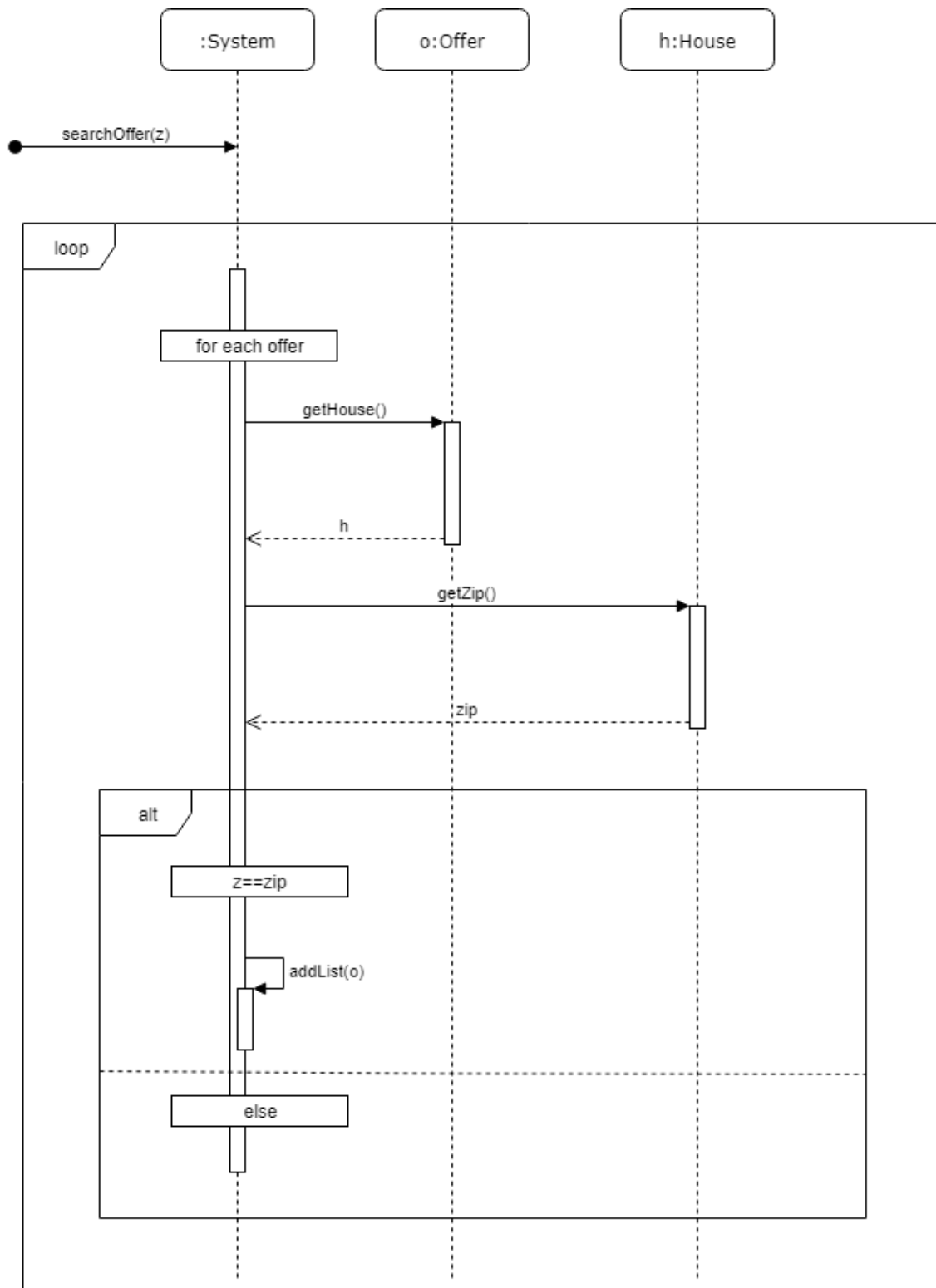
Comments to the diagrams:

- We decided to create an abstract class "User" whose sons are "Guest" and "Host" (these are the registered users). This can be a problem if we have someone who is a Guest and Host, but we think that these types of users can be registered two times.
- We decided to make another class for the registered user "Admin" who can only access to the non-approved offers.
- We decided to create an abstract class "Comment" whose sons are "Text" and "Numerical". We use this class to comment and rate offers and comments.
- We decided to make a class "Reserve" which includes a house, the guest who reserves the offer and initial date of the reserve.
- We decided to include the offers, the users, the admin and the houses in the system to have better access to all of them.
- We decided to include a modifyDate in the offer in order to know when to delete an offer
- We can conclude that our class diagram follows the patron "Composite" because we have lots of classes (user, comment, offer…) that composite the upper-class system. We also decided not to create a singleton class (System) at least until we code our application.

Miguel Ángel Sánchez
Juan Velasco
miguelangel.sanchezm@estudiante.uam.es
juan.velascoi@estudiante.uam.es

2. Sequence Diagrams

2.1 Buy Offer

Miguel Ángel Sánchez   *miguelangel.sanchezm@estudiante.uam.es*
Juan Velasco   *juan.velascoi@estudiante.uam.es*

2.2 Search Offer by Zip

Miguel Ángel Sánchez   *miguelangel.sanchezm@estudiante.uam.es*
Juan Velasco   *juan.velascoi@estudiante.uam.es*

2.3 Comments to the diagrams

- **buyOffer(g)**

In this diagram we have represented the method buyOffer, which is given the Guest g as a parameter.

The first thing we need to do is check if the offer is already reserved.

- If it is, we get the user that has reserved it and check if it is the same as the one trying to buy the offer.
  - If it is, we procced to the payment. We first get the credit card of the user and give it to the PaymenySystem which will return the Boolean paymentOk that says if the payment was successfully done.
    - If paymentOk is TRUE, we change the state of the offer to '2' (Which is the state that we have assigned as reserved)
    - If it's not, we ban the user changing its credit card. (We are not going to describe this method because it's not the chosen one)
  - If it's not the same user, the function needs to end because he can't buy that offer.
- If it's not reserved we directly proceed to the payment, the same way as we did before.
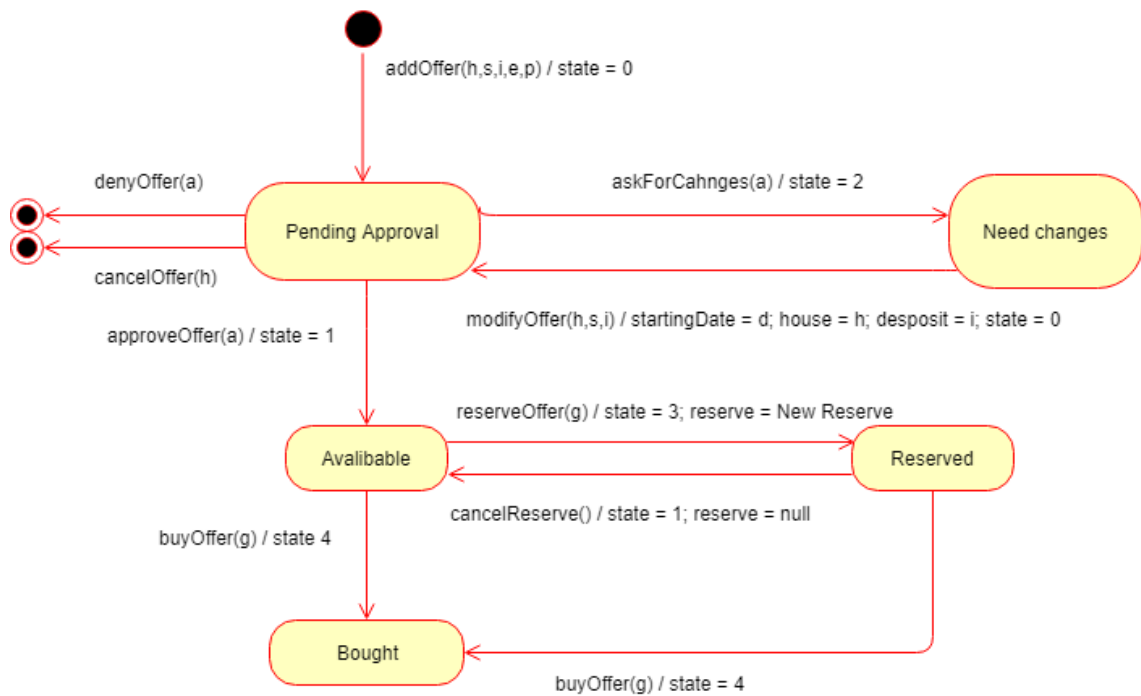
- **searchOffer(z)**

In this diagram is represented the method searchOffer(z), which is given the zip code as a parameter.

This method mainly has a loop that goes through all the offers and gets their houses and, from those houses, the zips. Then it checks if that zip is the same as the one given as a parameter. If it is, what we have done is add that offer to a list, which will be the one that is finally returned as the result of the search.

Miguel Ángel Sánchez   miguelangel.sanchezm@estudiante.uam.es
Juan Velasco   juan.velascoi@estudiante.uam.es

3. State transition diagrams

3.1 Class: Offer

3.2 Class: User

Miguel Ángel Sánchez
Juan Velasco

miguelangel.sanchezm@estudiante.uam.es
juan.velascoi@estudiante.uam.es
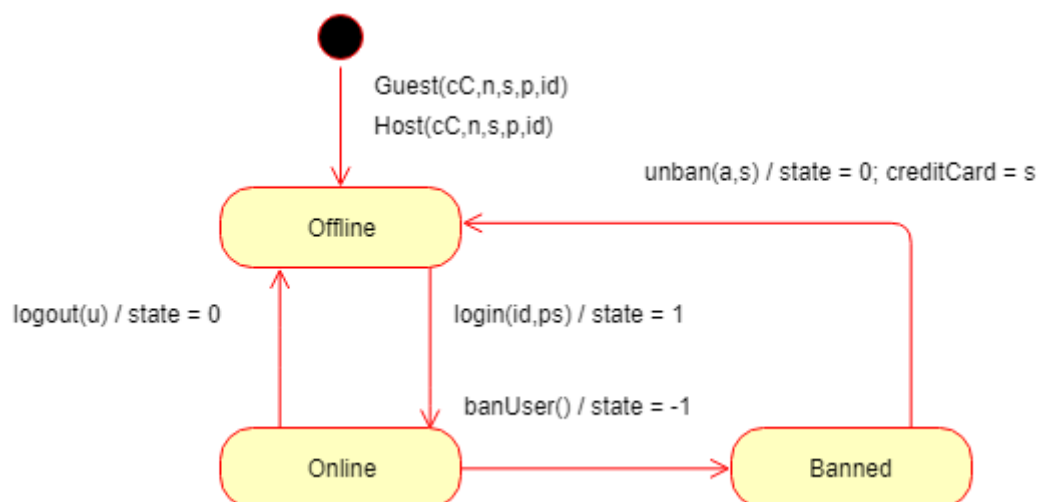
3.3 Comments to the diagrams

- **Offer:**

In this state diagram, when we create an offer, it becomes pending of approval. Then, the admin can ask for changes and the user has to change the offer. When the offer is in pending of approval, the admin can also deny the offer (deleting the object) or approve it and the host can cancel it (deleting the object).

When an offer is available, a guest can reserve an offer or buy it. When it is reserved, if the guest doesn't buy it, it changes to available again.

Finally, the offer changes its state to bought and it is added to the register of the application (so it stays there).

- **User:**

First, we decided that a registered user can't be deleted from the system. When a user is created it becomes offline. Once the user logins into the system, the user appears as online. If the user is a guest and introduces a false credit card, he gets banned and he has to wait for an admin to unban him (changing his state to offline). When a user finishes, he logout and change his state to offline again.

Miguel Ángel Sánchez
Juan Velasco

miguelangel.sanchezm@estudiante.uam.es
juan.velascoi@estudiante.uam.es

4. Traceability Matrix

	Methods					
Requirements	searchOfferType	SearchOfferZip	SearchOfferDate	login	logout	searchOfferRate
Search offers by type	X					
Search offers by zip code		X				
Search offers by date			X			
Login				X		
Logout					X	
Search offers by rate						X
Search and see reserved offers						
Search and see bought offers						
Introduce house						
Introduce offer						
Cancel offer						
Modify offer						
See own offers						
Reserve offer						
Buy offer						
Comment offer or comment						
Rate offer or comment						
Change credit card						
Approve offer						
Deny offer						
Ask for changes in offer						

Miguel Ángel Sánchez
Juan Velasco
miguelangel.sanchezm@estudiante.uam.es
juan.velascoi@estudiante.uam.es

Methods

Requirements	searchOfferReserved	searchOfferBought	addHouse	addOffer	unban	banUser
Search offers by type						
Search offers by zip code						
Search offers by date						
Login						
Logout						
Search offers by rate						
Search and see reserved offers	X					
Search and see bought offers		X				
Introduce house			X			
Introduce offer				X		
Cancel offer						
Modify offer						
See own offers						
Reserve offer						
Buy offer						
Comment offer or comment						
Rate offer or comment						
Change credit card					X	X
Approve offer						
Deny offer						
Ask for changes in offer						

Methods

Requirements	seeOffers	cancelReserve	commentComment	rateComment	approveOffer	denyOffer
Search offers by type						
Search offers by zip code						
Search offers by date						
Login						
Logout						
Search offers by rate						
Search and see reserved offers						
Search and see bought offers						
Introduce house						
Introduce offer						
Cancel offer						
Modify offer						
See own offers	X					
Reserve offer						
Buy offer						
Comment offer or comment			X			
Rate offer or comment				X		
Change credit card						
Approve offer					X	
Deny offer						X
Ask for changes in offer						

10

Miguel Ángel Sánchez
Juan Velasco

miguelangel.sanchezm@estudiante.uam.es
juan.velascoi@estudiante.uam.es

Requirements / **Methods**	askForChanges	cancellOffer	modifyOffer	reserveOffer	buyOffer	commentOffer	rateOffer
Search offers by type							
Search offers by zip code							
Search offers by date							
Login							
Logout							
Search offers by rate							
Search and see reserved offers							
Search and see bought offers							
Introduce house							
Introduce offer							
Cancel offer		X					
Modify offer			X				
See own offers							
Reserve offer				X			
Buy offer					X		
Comment offer or comment						X	
Rate offer or comment							X
Change credit card							
Approve offer							
Deny offer							
Ask for changes in offer	X						