# Scopo del documento

Il presente documento riporta la definizione dell’architettura del progetto Skupply usando diagrammi delle classi in UML e codice OCL. Nel precente documento è stato presentato il diagramma degli use case, il diagramma di contesto e quello dei componenti. Ora, tenendo conto dei diagrammi appena indicati, viene definita l’architettura del sistema dettagliando da un lato le classi che dovranno essere implementate a livello di codice e dall’altro la logica che regola il comportamento del software.

Le classi vengono rappresentate tramite un diagramma delle classi in linguaggio UML mentre la logica, viene descritta in OCL.

# Diagramma delle classi

Nel presente capitolo vengono riportate le classi previste nell’ambito del progetto Skupply.

Ogni componente presente nel diagramma dei componenti diventa una o più classi. Tutte le classi individuate sono caratterizzate da un nome, una lista di attributi che identificano i dati gestiti dalla classe e il loro tipo ed infine una lista di metodi che definiscono le operazioni previste all’interno della classe.

Ogni classe può essere anche associata ad altre classi tramite relazioni di ereditarietà e/o di semplici relazioni. Queste ultime vengono indicate tramite delle linee direzionate e no, un’etichetta che indica in sintesi la relazione e opzionalmente la cardinalità.

Vengono riportate di seguito le classi individuate a partire dal diagramma di cotesto e dei componenti.

## Gestione utenti

Analizzando il diagramma di contesto realizzato si nota la presenza di due attori “acquirente” e “venditore”.

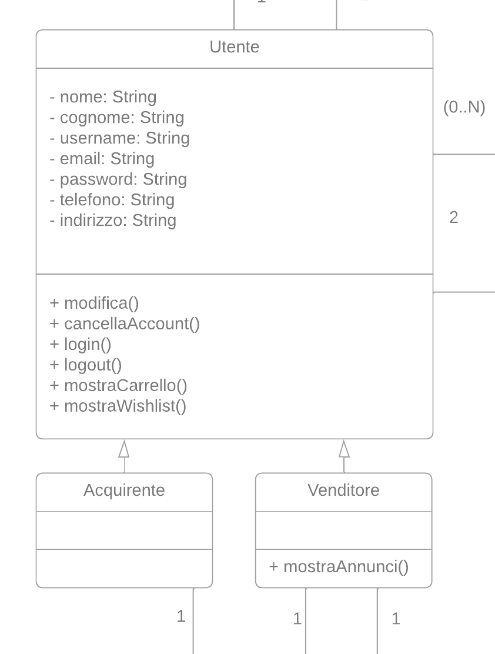
Entrambi gli attori possono essere visti come entità che ereditano da un unico attore “utente”, il quale detiene le informazioni base che sono comuni a tutti e due gli attori appena citati.

I due attori di distinguono per il loro ruolo all’interno del sistema. L’attore “acquirente” è colui che utilizza l’applicazione per visualizzare ed acquistare gli articoli presenti nel negozio, i quali vengono creati, pubblicati ed eventualmente poter essere rivisti dall’attore “venditore”.

Per questo motivo sono state individuate tre classi: Utente, Acquirente e Venditore.

Le classi Acquirente e Venditore hanno attributi e funzioni in comune con Utente tramite la generalizzazione.

Da sottolineare che la classe Utente implementa le interfacce del componente Gestore profilo del diagramma dei componenti.

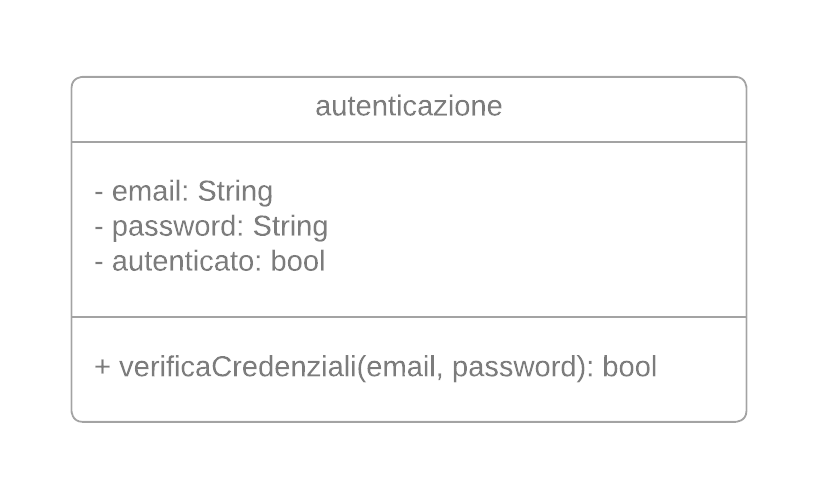


## Gestione autenticazione

Dal diagramma dei componenti viene individuato il componente “Gestore autenticazione” il quale si occupa delle fasi di autenticazione, registrazione dell’utente e dell’eventuale cancellazione del profilo.

Per questo motivo è stata individuata una classe Autenticazione che si occuperà delle funzionalità appena descritte.

La funzione verificaCredenziali() si occupa, tramite il passaggio alla classe degli attributi email e password, di verificare le credenziali inserite e in caso di conferma, cambiare lo stato dell’utente in autenticato.



## Gestione pagamento e ordine

Dal diagramma di contesto e dei componenti vengono individuati più entità che permettono la gestione dei pagamenti e degli ordini, i quali si interfacciano anche a sistemi esterni.

Da queste entità viene individuata la seguente classe: Ordine.

Tale classe oltre che a modificare le quantità degli articoli disponibili all’interno del negozio in seguito all’acquisto, permette l’effettuazione del pagamento tramite interfaccia al sistema esterno per i pagamenti Paypal e anche alla creazione del codice di tracking, se richiesto, sempre tramite interfaccia al sistema esterno per le spedizioni Shippo (rif. Diagramma di contesto).

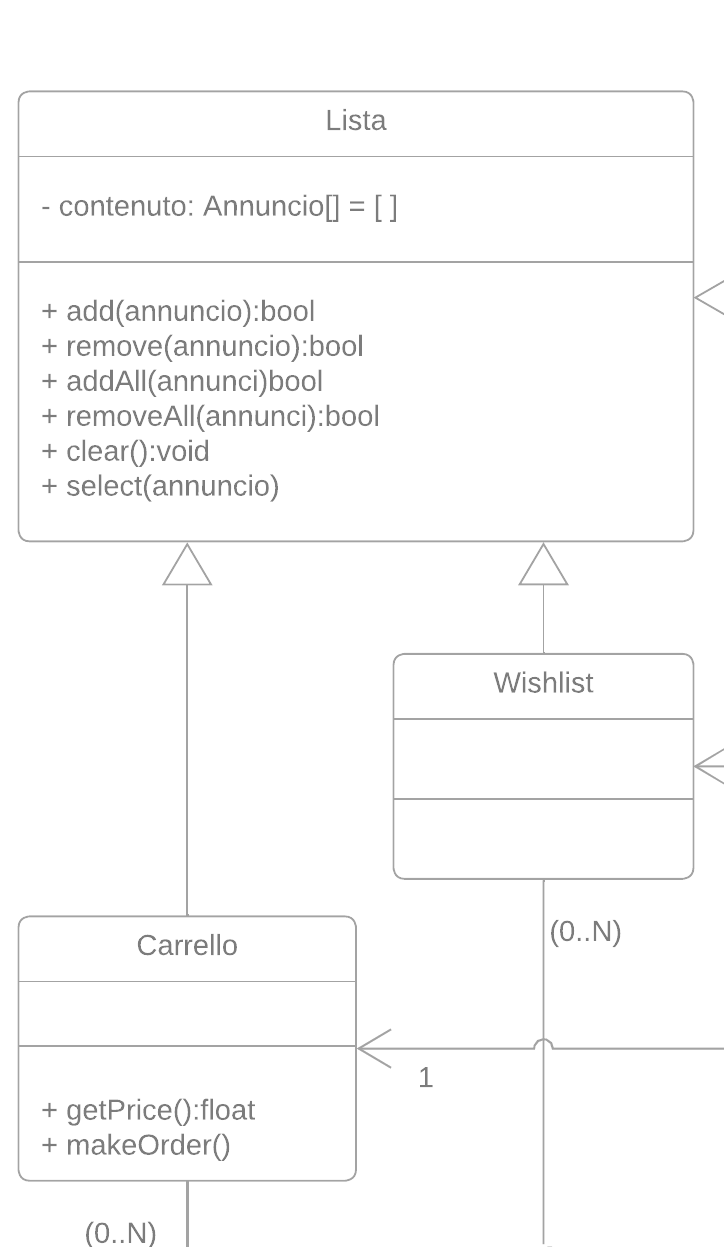
Immagine che contiene testo

Descrizione generata automaticamente

## Gestione carrello e wishlist

Dal diagramma dei componenti vengono individuate le classi Carrello e Wishlist le quali permettono la gestione di una lista di contenuti, in questo caso annunci.

In quanto entrambe le classi presentano molte similitudini ad eccezione del carrello che presenta le funzionalità per effettuare il checkout, le classi presentano ereditarietà a partire dalla classe Lista che gestisce la sola lista dei contenuti e quindi mette a disposizione le funzionalità di aggiunta, rimozione e cancellazione dei contenuti dalla lista.

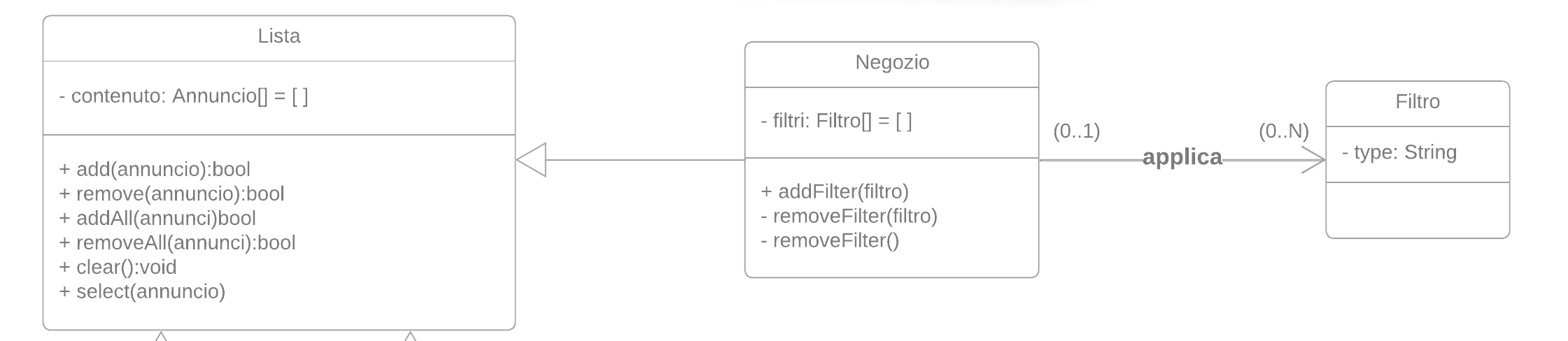


## Gestione negozio

Dal diagramma dei componenti viene individuata la classe Negozio, la quale gestisce la visualizzazione degli annunci in base ai termini di ricerca inseriti dall’utente e degli eventuali filtri utilizzati.

In modo analogo alle classi precedentemente descritte (Carrello e Wishlist), la classe Carello eredita gli attributi e le funzioni della classe Lista.

Inoltre, come indicato ad inizio punto, la classe Negozio deve poter anche applicare i filtri ai termini di ricerca. Per questo motivo Negozio, tramite una relazione “applica”, è relazionata alla classe Filtro, derivata dal diagramma dei componenti.



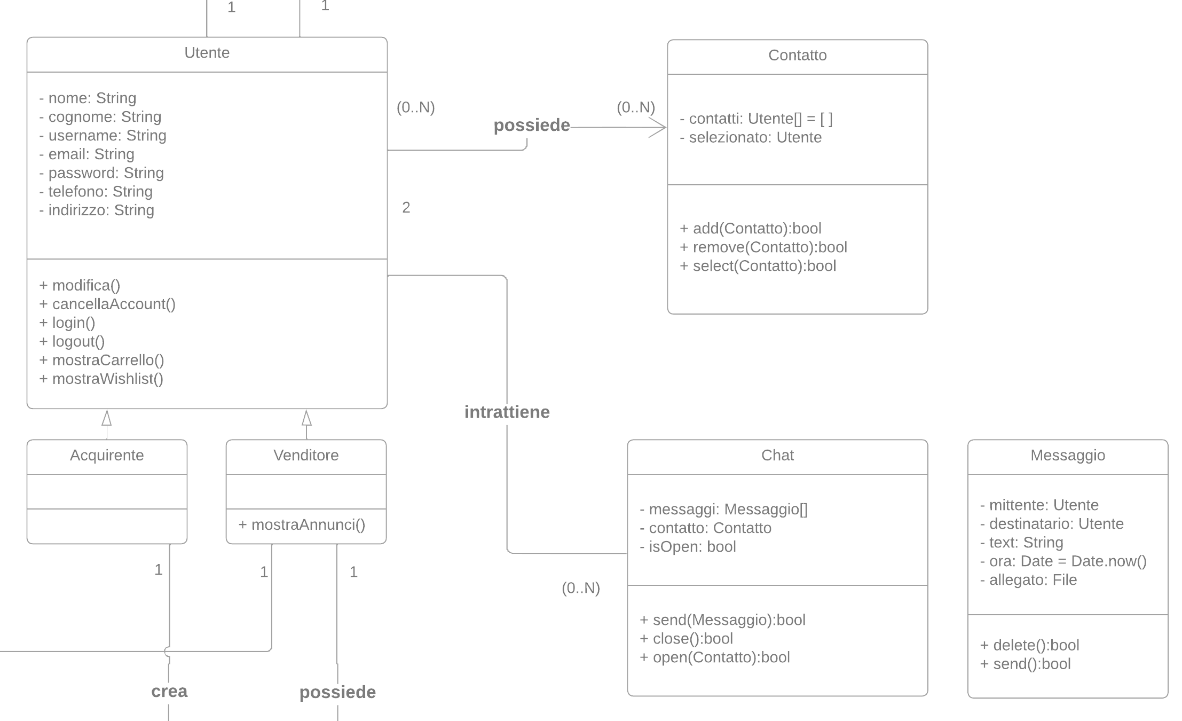
## Gestione chat

Dal diagramma dei componenti e dal diagramma di contesto, viene individuata la necessità di dover creare delle classi per poter permettere la possibilità di scambiare messaggi e allegati fra i vari utenti all’interno della chat.

Si vengono a formare le seguenti classi: Contatto, Messaggio e Chat. In ordine, la classe Contatto non è altro che una classe che implementa le funzionalità di una rubrica; la classe Messaggio che rappresenta l’entità che viene scambiata fra gli utenti ed infine la classe Chat che gestisce lo scambio di messaggio fra due utenti.

Nessuna delle classi presenta ereditarietà ma solamente relazioni semplici.

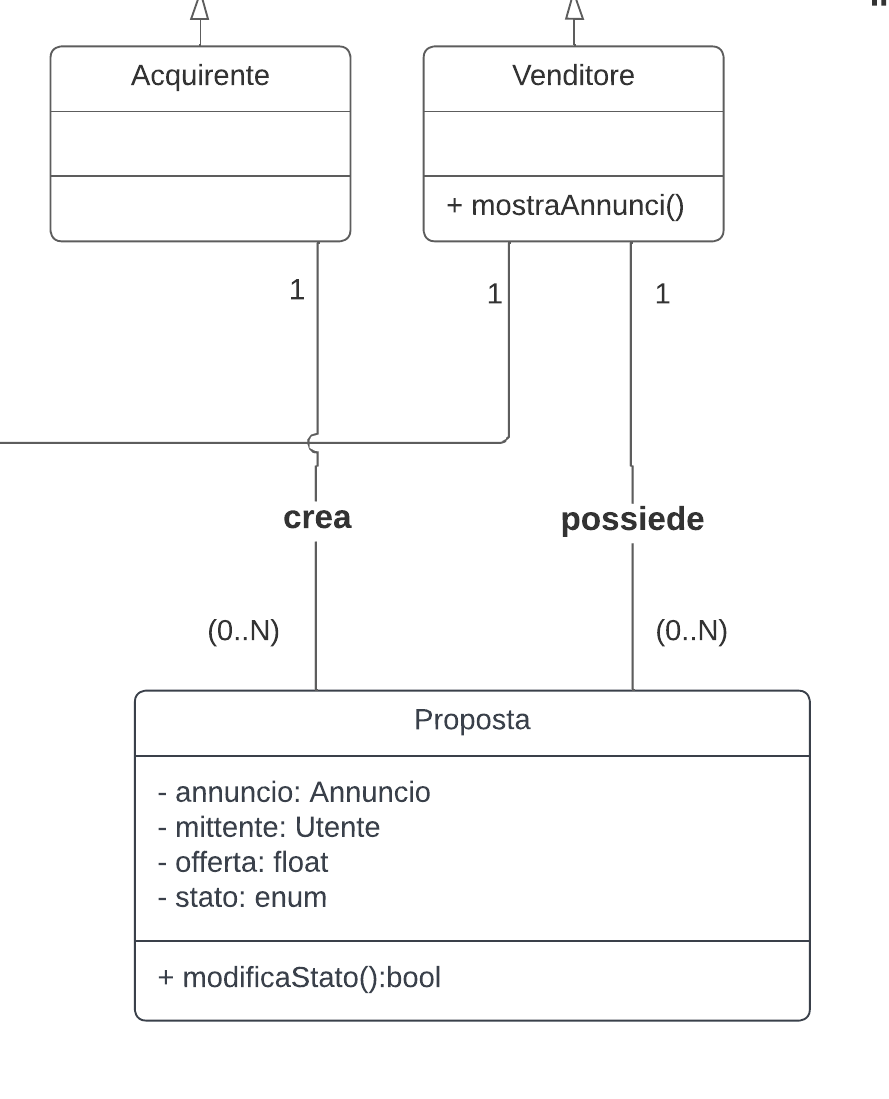
Di seguito, le classi con i relativi attributi e le funzioni messe a disposizione.



## Gestione proposte

Dal diagramma dei componenti si nota la necessità di creare una classe che permetta la gestione delle proposte d’acquisto ovvero, la possibilità di creazione ed accettazione e/o rigetto di esse da parte dei venditori.

Per tali motivi viene creata la classe Proposta, la quale condivide due relazioni con la classe Acquirente e Venditore in quanto entrambi gli utenti, anche se hanno funzionalità diverse, devono agire sulla stessa entità.



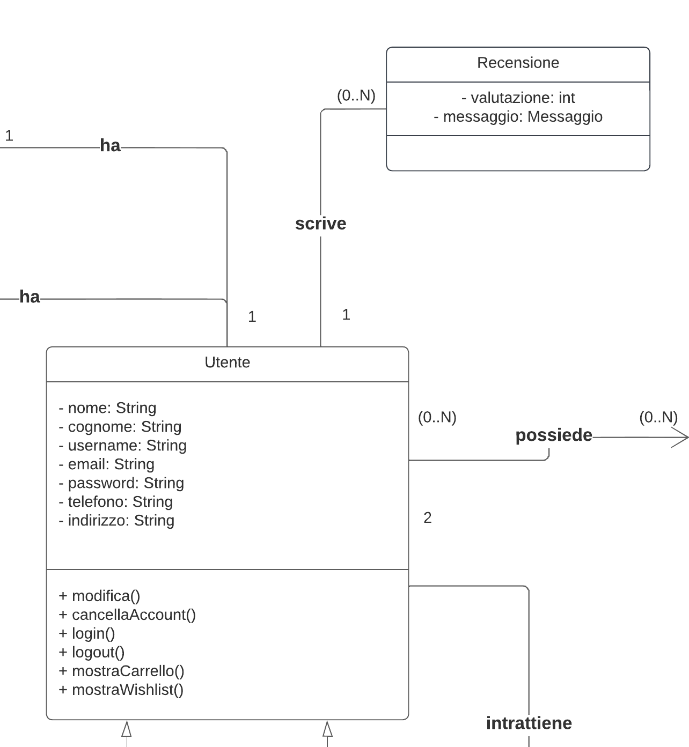
(stranamente immagine buggata)

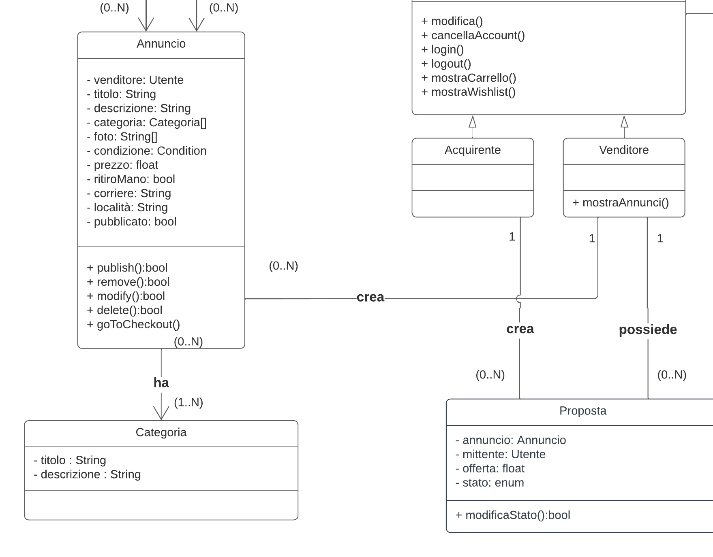
## Gestione e visualizzazione annunci

Dal diagramma dei componenti e stata individuata la classe Annuncio che rappresenta l’entità più importante del sistema Skupply. Tale classe detiene tutti gli attributi necessari per la rappresentazione più dettagliata possibile dell’annuncio e mette a disposizione le funzionalità denotate sia dalle interfacce del componente Gestore annunci e Visualizzatore annuncio ma anche dei flussi del diagramma di contesto.

Vista l’importanza della classe e delle funzionalità richiesta, si è optato per una divisione delle informazioni rendendole classi assestanti. Categoria e Recensione sono un esempio.

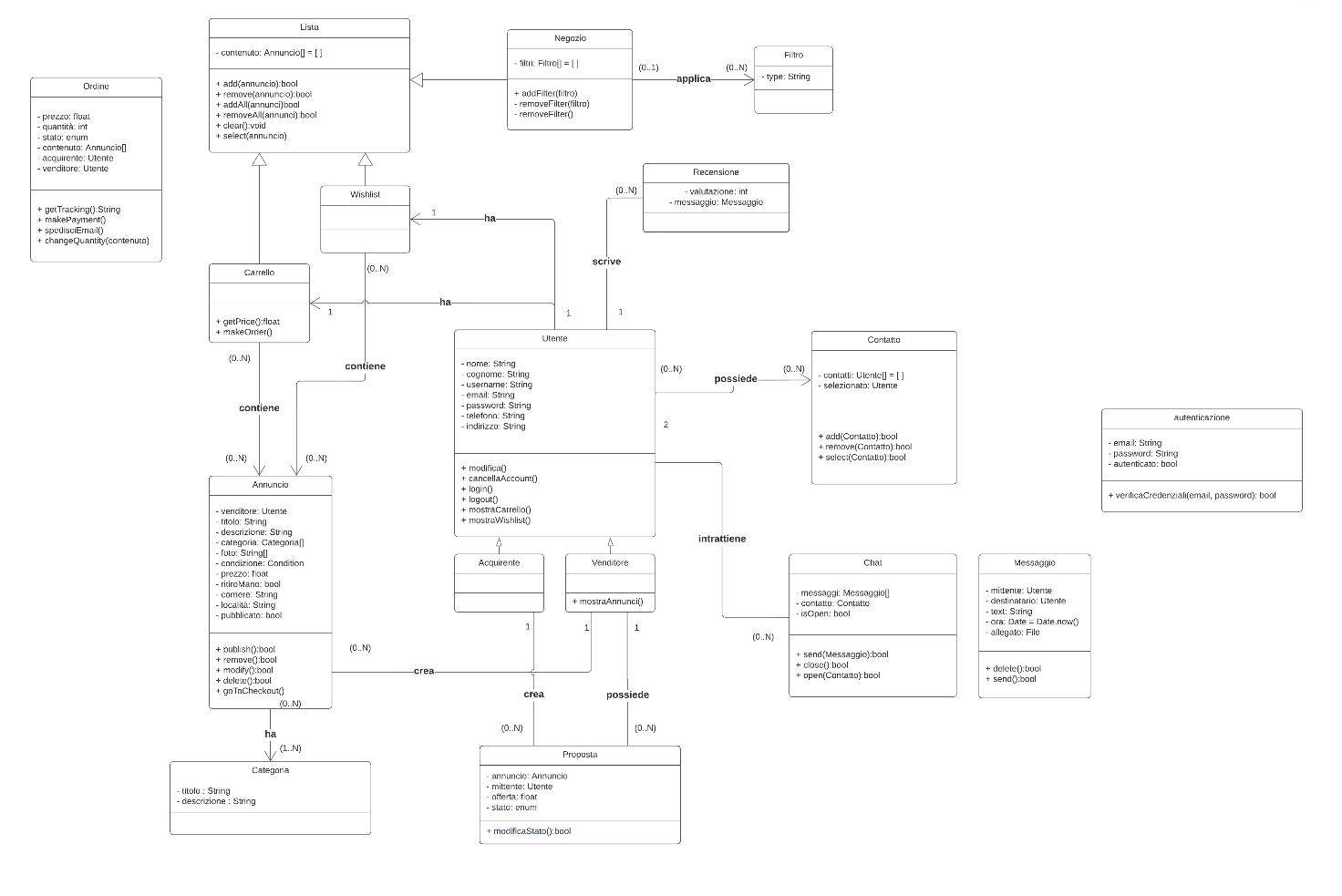
Le tre classi risultanti sono in relazione oltre che tra di loro, anche alle classi Utente e Venditore.





(di nuovo bug immagini)

## Diagramma delle classi complessivo



## Codice in Object Constraint Language (OCL)