



# UNIVERSE

---

Progetto ideato dagli studenti dell'[università di Trento](#)

[Lorenzo Masè](#)

[Lorenzo Zanini](#)

[Pietro Bassa](#)

## Indice

1. [Diagramma delle classi](#)
  - 1.1. [Sistema autenticazione](#)
  - 1.2. [Pagina “SignUp/Login”](#)
  - 1.3. [Gestione visualizzazione Università preferite](#)
  - 1.4. [Gestione Università](#)
  - 1.5. [Gestione chat](#)
  - 1.6. [Pagina “Ricerca”](#)
  - 1.7. [Diagramma delle classi complessivo](#)
2. [Codice in Object Constraint Language](#)
  - 2.1. [Autenticazione](#)
  - 2.2. [Messaggistica](#)
  - 2.3. [Messaggistica per Tutor](#)
  - 2.4. [Ricerca Università](#)
  - 2.5. [Aggiungi Università](#)
3. [Diagramma delle classi con codice OCL](#)

## **Scopo del documento**

Il seguente documento contiene la definizione dell'architettura del progetto UNInverse usando diagrammi delle classi in UML(Unified Modeling Language) e codice in OCL(Object Constraint Language).

Dopo aver descritto la struttura del sistema nel precedente documento, seguendo la medesima progettazione si descrivono le varie classi e la logica dietro il software.

# Diagramma delle classi

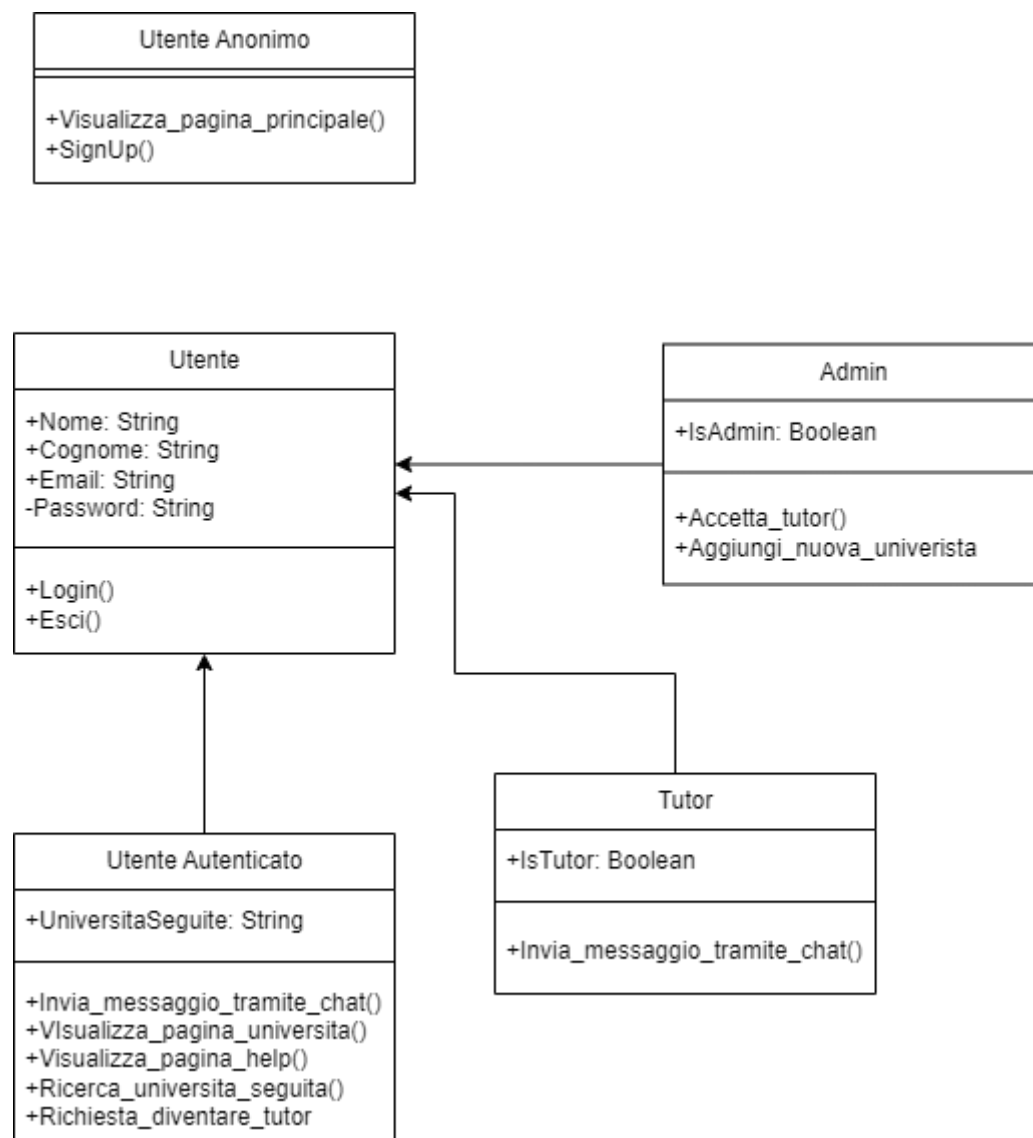
## Sistema autenticazione

Gli utenti possono essere di 4 tipologie: Utente Anonimo, Utente Autenticato, Admin e Tutor.

La prima classe cioè quella “Utente” comprenderà le ultime tre della lista.

Ogni componente sarà quindi parte di una classe corrispondente, queste conterranno gli attributi e le funzioni.

Questa gestione permetterà di definire sia la componente individuale sia il rapporto tra le singole, poiché le funzioni di una classe spesso entreranno in contatto con altre classi, che richiederanno informazioni.

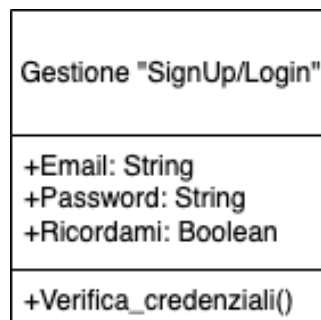


## **Pagina “SignUp/Login”**

Questa classe rappresenta la pagina che permette l'autenticazione, gli attributi saranno richiesti all'utente, che una volta averli inseriti, e richiesto il SignUp/Login farà partire la funzione Verifica\_credenziali().

La funzione non farà altro che controllare se le credenziali sono presenti nel database o se devono essere inserite.

Questo comporta un primo rapporto con un'altra componente del sito (database).

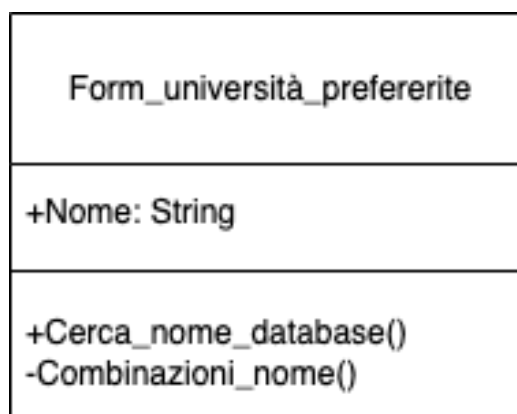


## **Gestione visualizzazione Università preferite**

Questa classe viene definita analizzando la pagina Home, questa ha il compito di controllare la visualizzazione delle università una volta inserite tra le preferite.

Le funzioni di ricerca permettono di gestire la visualizzazione.

Ci sarà la possibilità di ricercare sia per corso di studi sia per Ateneo.

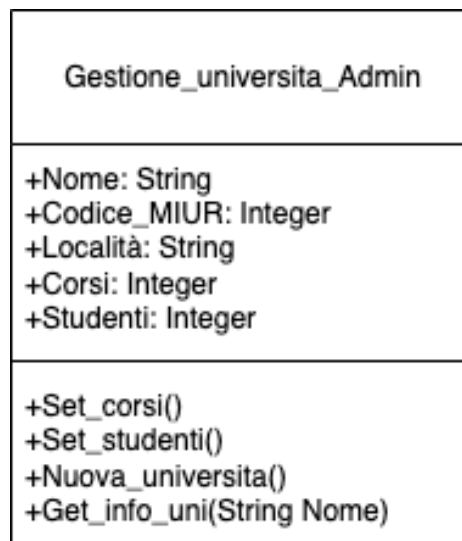


## **Gestione Università**

La seguente classe ha il ruolo di gestire le Università all'interno del sito, infatti UNInverse ha un rapporto molto stretto con le Università.

Gli admin dovranno anche contattare queste ultime per controllare l'affidabilità delle richieste fatte dagli aspiranti tutor.

Le funzioni principali di questa classe saranno di ottenere e inserire nel sito i dati delle varie università.



## **Gestione chat**

La classe gestione chat avrà il ruolo di controllare il flusso dei messaggi tra gli utenti, questa lavorerà a stretto contatto con il database poiché questo farà da contenitore per le chat.

Gli attributi saranno formati dalla dimensione dei messaggi e il loro contenuto, mentre le funzioni permetteranno l'invio e la ricezione.



## **Pagina “Ricerca”**

La pagina di Ricerca conterrà tutte le Università presenti nel database e attraverso un algoritmo di ricerca permetterà di filtrare tra queste.

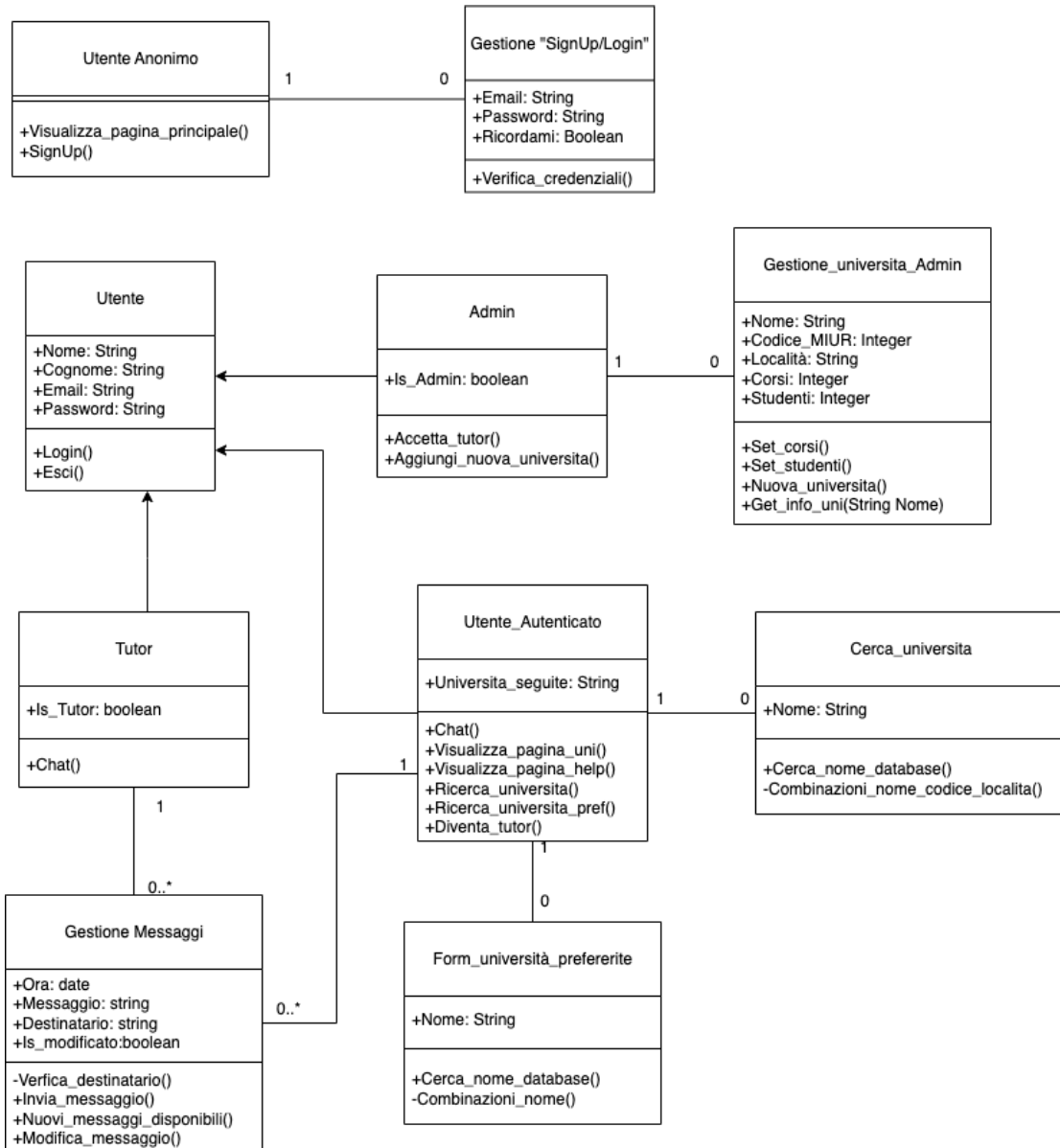
Le funzioni saranno principalmente relative alle università che saranno già state inserite interamente nel database in precedenza.

Cerca_universita
+Nome: String
+Cerca_nome_database() -Combinazioni_nome_codice_localita()

## Diagramma delle classi complessivo

Inseriamo successivamente il diagramma delle classi descritte e definite finora.

Queste verranno poi descritte in OCL.





## Codice in Object Constraint Language

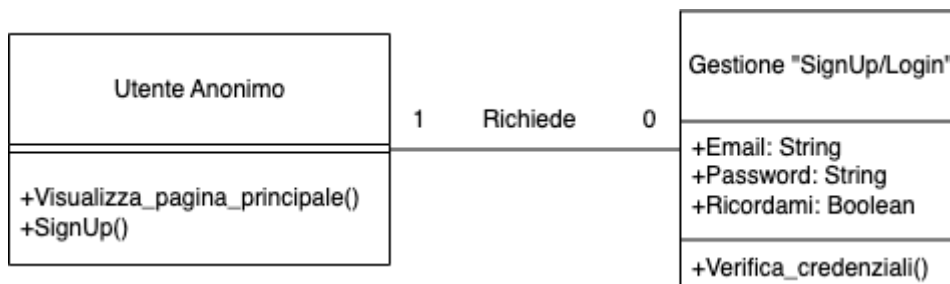
Questa parte del documento descriverà come le varie classi comunicheranno attraverso attributi e dati in comune.

Molte delle classi descritte infatti avranno bisogno di dati elaborati da altre precedentemente.

Questi rapporti verranno descritti in **OCL**.

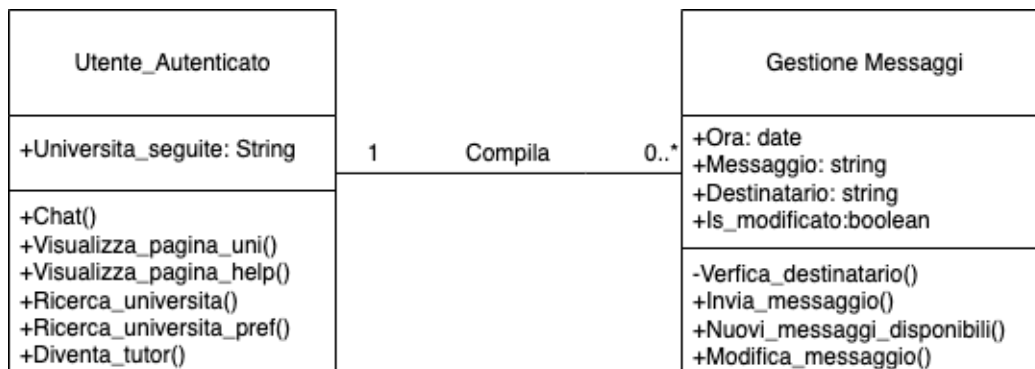
### Autenticazione

L'utente non ancora autenticato potrà richiedere, tramite la compilazione di una form, di accedere nella sezione dedicata.



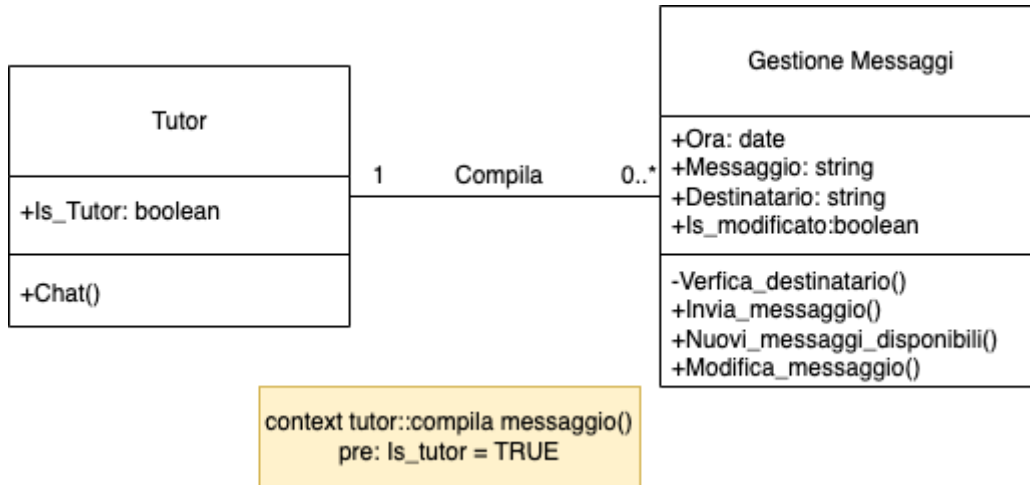
### Messaggistica

L'utente può compilare dei messaggi da inviare al tutor



## Messaggistica per Tutor

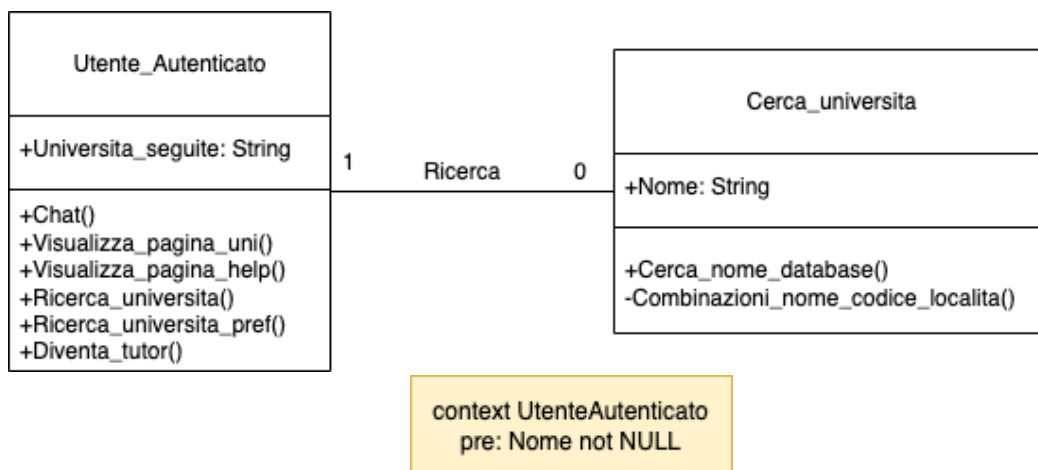
Il tutor per inviare messaggi dovrà avere la conferma di essere un tutor. Quindi il campo `Is_Tutor` dovrà essere impostato su *True*



## Ricerca Università

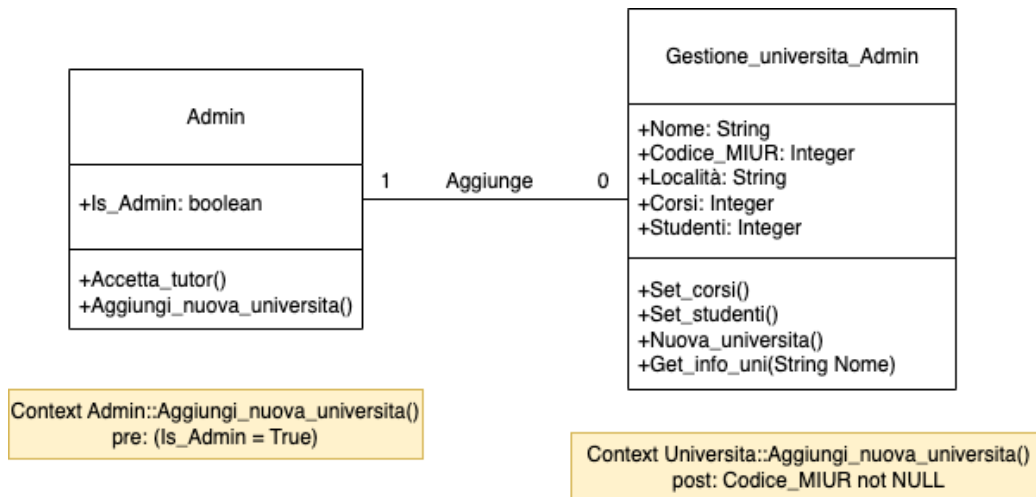
L'utente autenticato potrà eseguire una ricerca, sul database, tra le università.

Stessa condizione verrà applicata anche per la ricerca tra le università preferite.



## Aggiungi Università

Un admin ha la possibilità di aggiungere una nuova università nel database. Per poterlo fare dovrà avere l'attributo `Is_Admin` impostato a `True`. Quando si aggiunge un università, l'admin dovrà controllare che il valore del codice MIUR sia impostato. Nel caso non lo fosse non sarà considerata valida l'operazione.



# Diagramma delle classi con codice OCL

