Unireview

Architettura del progetto

Indice

SCOPO DEL DOCUMENTO

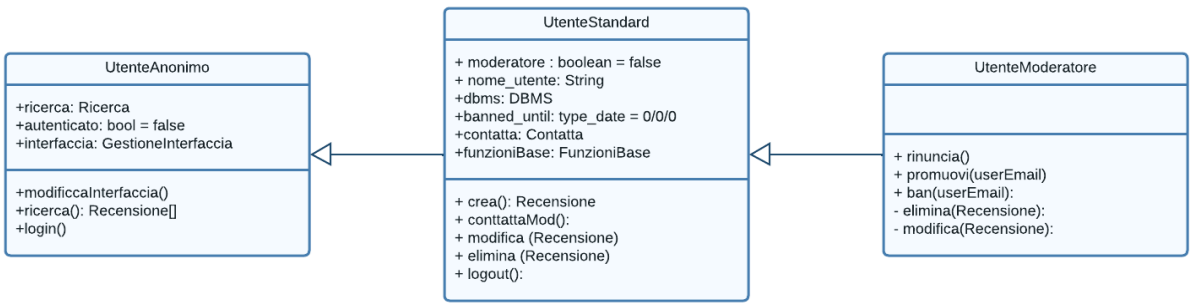
Nel documento verrà presentata la struttura dell'architettura della webapp UniReview nella definizione delle sue classi attraverso diagrammi in linguaggio UML, così come codice OCL per la definizione della logica del sistema. Questo documento si basa sulle specifiche del documento precedente, in particolar modo in riferimento agli use-case diagram e al diagramma dei componenti. L'architettura del sistema fa riferimento alla progettazione già presentata, e in seguito questa dovrà poi essere tradotta a livello di codice tenendo a mente la logica già definita riguardo al comportamento del software.

1. DIAGRAMMA DELLE CLASSI

In questo capitolo verranno mostrate le classi da implementare nel progetto UniReview. Per permettere l'implementazione delle specifiche già descritte nei documenti precedenti. Ogni componente si traduce in una o molteplici classi in riferimento anche ai requisiti funzionali. Questo si nota facilmente osservando il nome della classe, gli attributi che contiene, così come i metodi volti alla modifica e controllo di questi.

Qui sotto si presentano le classi previste per il progetto individuate nei diagrammi di contesto e dei componenti.

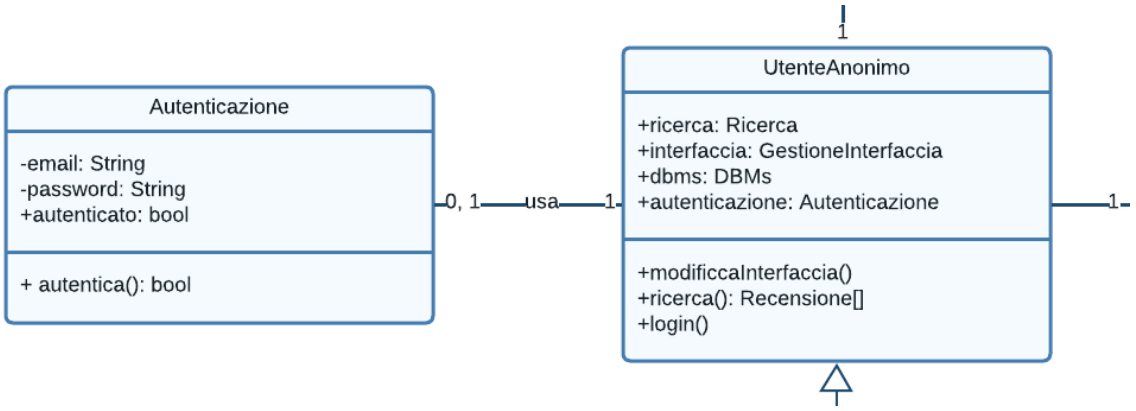
1.1 Utenti e relative funzionalità



Come mostrato dal diagramma dei componenti, si nota la presenza di tre attori: Utente anonimo, Utente standard e Utente moderatore. Come viene

definito dagli use-case diagram, questi hanno funzionalità condivise in relazione IS-A. La classe UtenteAnonimo racchiude tutte le funzionalità base del sistema, a cui tutti hanno accesso. Ogni classe con gerarchia superiore avrà accesso a funzionalità ulteriori, come definito dal RF2.

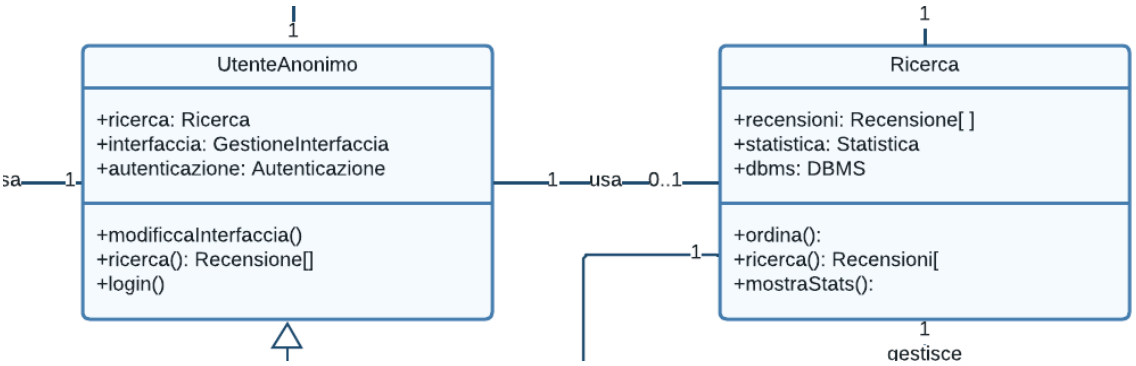
1.2 Sistema di autenticazione



Come osserviamo dagli use-case diagram, all'utente anonimo viene data la possibilità di autenticarsi. Per rendere ciò possibile la classe UtenteAnonimo possiede un riferimento alla classe Autenticazione, ovvero un modulo da compilare che si assicura che l'utente sia idoneo all'autenticazione. Per assicurarsi la privacy dell'utente, come si osserva anche dai RNF, la classe non memorizza le credenziali inserite, appoggiandosi alle API di ateneo per effettuare la conferma delle credenziali.

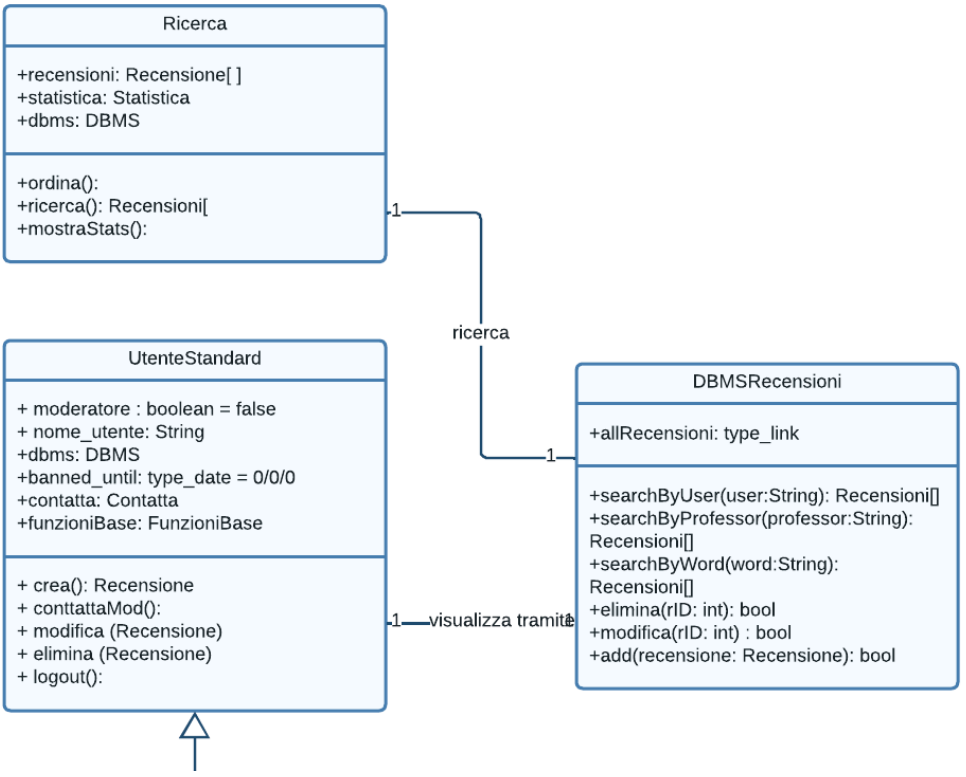
La classe Autenticazione è inoltre responsabile della verifica che l'utente che sta tentando di effettuare l'accesso non risulti "sospeso", in questo caso verrà negato il suo accesso.

1.3 Ricerca



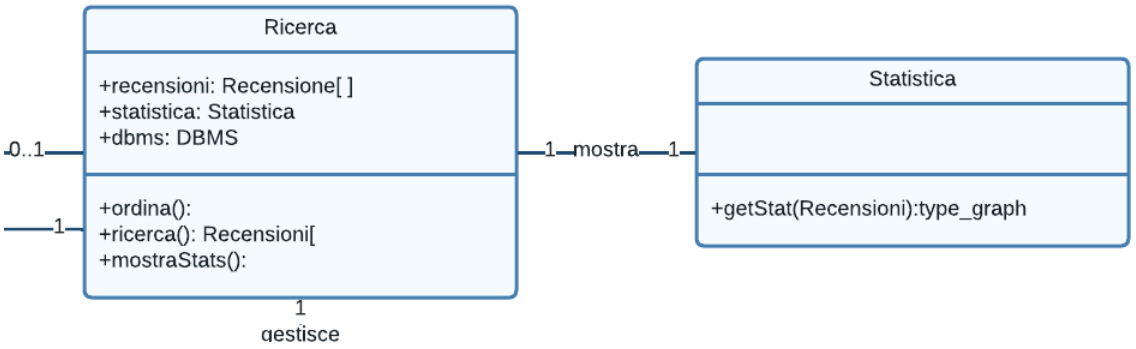
La classe Ricerca, come si osserva dal diagramma di contesto e dei componenti, si interfaccia con l'utente per permettere la ricerca di recensioni all'interno del database. La classe UtenteAnonimo si appoggia infatti su di essa per poter consultare il database della webapp al momento della ricerca. Ad essa vengono anche date le funzionalità di ordinamento definite nel documento dei requisiti funzionali.

1.4 DBMSRecensioni



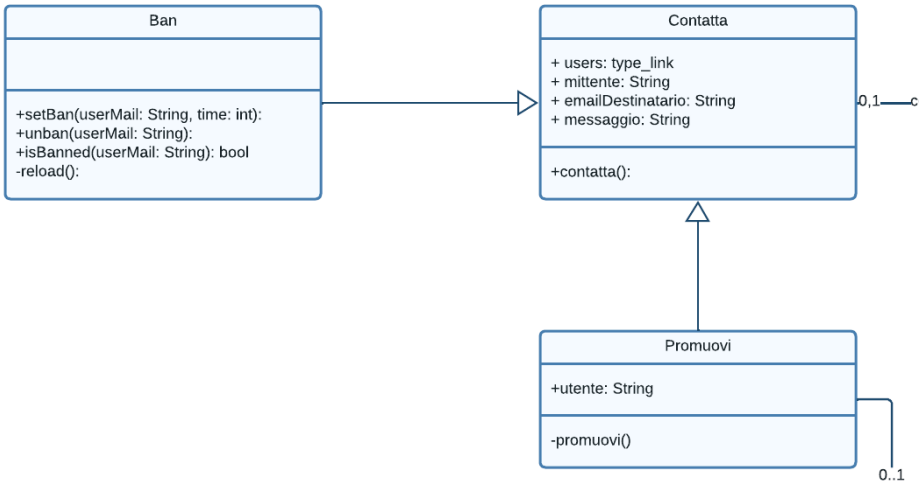
Questa classe è a stretto contatto con il database di UniReview, presente in relazione 1..1 con le classi UtenteStandard e Ricerca. La classe sta alla base delle funzioni di ricerca, aggiunta, modifica e rimozione di recensioni all'interno del database, come si osserva anche dal diagramma dei componenti. La classe contiene, infatti, un riferimento al link del database della webapp che viene consultato per qualsiasi operazione di ricerca o modifica.

1.5 Statistiche



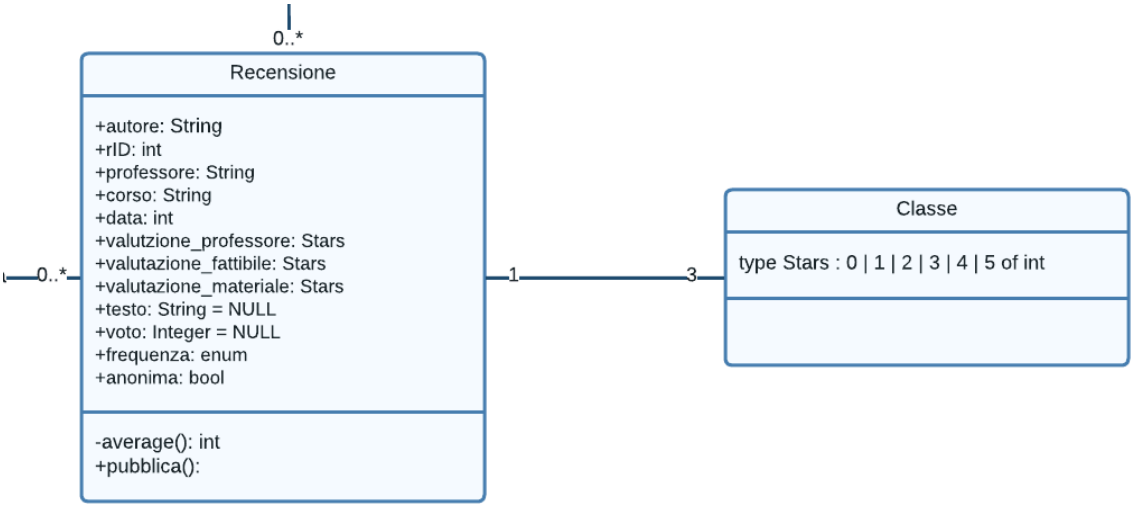
La classe si appoggia a Ricerca per fornire tutte le statistiche che compaiono in seguito alla ricerca delle recensioni, come viene descritto anche dagli use-case diagram.

1.6 Classi di invio di messaggi



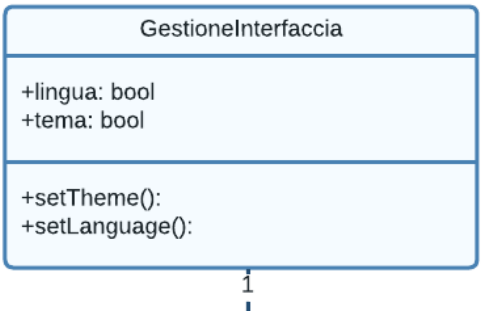
Dagli use-case diagram e dal diagramma dei componenti si nota che ci sono diversi attori che svolgono azioni che comportano l'invio di una notifica email tramite API esterne. Questo meccanismo viene gestito dalla classe Contatta, strutturata come un modulo da compilare per permettere a UtenteStandard di contattare utenti moderatori inserendo un messaggio. Ci sono poi due classi in relazione IS-A: la classe Promuovi permette ai moderatori di promuovere utenti standard a moderatori, scatenando quindi l'invio del messaggio di upgrade all'utente interessato. La classe Ban permette ad UtenteModerarore di sospendere temporaneamente l'accesso ad un utente sulla piattaforma. Anche in questo caso l'utilizzo delle funzionalità della classe scatenano l'invio di un'email di notifica.

1.7 Recensioni



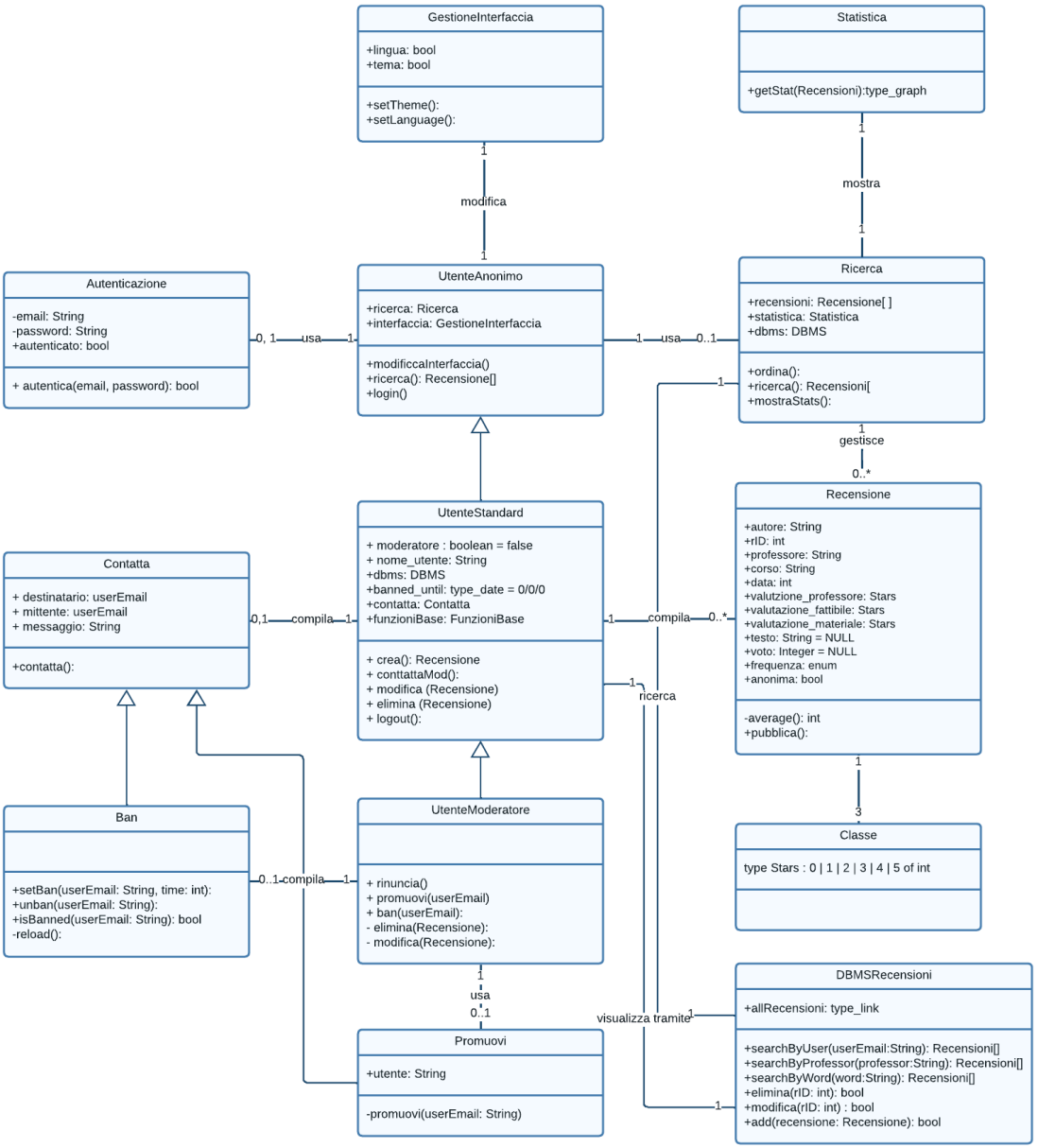
La classe Recensione è l'oggetto alla base di tutte le transazioni che avvengono tra il sistema della webapp e il relativo database. Si presenta come un modulo di cui compilare i campi, specificati dai RF. La classe presenta anche un metodo per calcolare la media delle tre valutazioni date, descritte attraverso il tipo specifico Stars, ovvero un numero intero che va da 0 a 5 compresi. Alla classe viene anche dato un ID per permetterne una ricerca più efficace.

1.8 Gestione interfaccia



Questa classe, come si nota dal diagramma di contesto, permette all'utente di modificare le preferenze dell'interfaccia grafica quali la lingua di sistema e il tema attraverso appositi metodi.

1.9 Diagramma complessivo

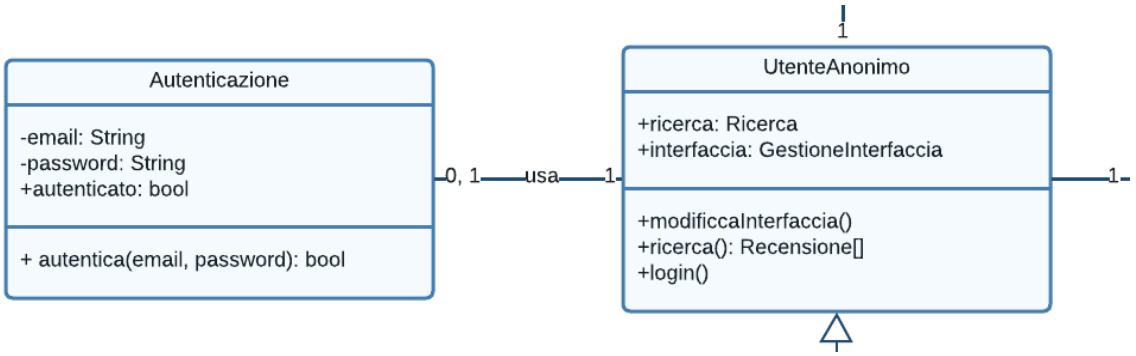


Qui sopra viene mostrato il diagramma complessivo delle classi del progetto UniReview, con relative relazioni di eredità e dipendenza.

2. CODICE IN OBJECT CONSTRAINT LANGUAGE

Nel seguente capitolo verrà descritta la logica dietro le classi e i metodi del progetto UniReview. Questa verrà mostrata attraverso l'utilizzo di codice in OCL in modo tale da chiarire meglio il contenuto dei diagrammi UML.

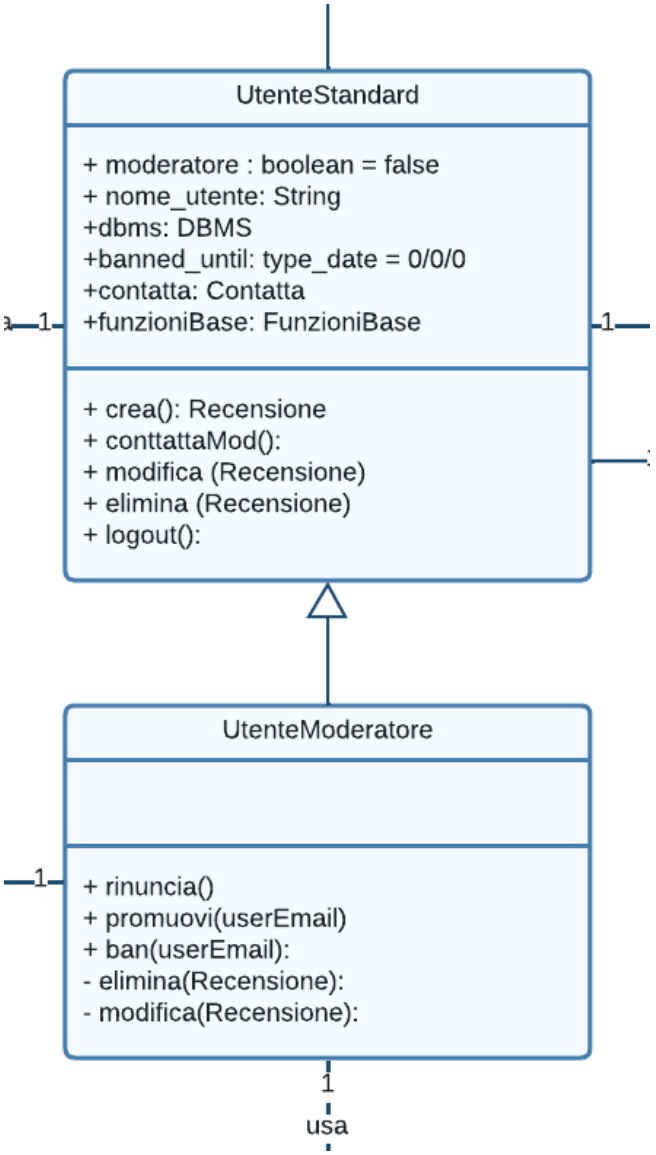
2.1 Autenticazione




```
context UtenteAnonimo :: login()
pre : UtenteAnonimo.autenticato = false
post: UtenteStandard.autenticato = true
      UtenteStandard.moderatore = _
```

Alla classe UtenteAnonimo viene data una flag "autenticato" che segna se l'utente abbia effettivamente effettuato il login. Una volta completato il processo di autenticazione, la flag viene impostata a true e l'utente non può quindi più autenticarsi. Effettuato il login si consulta anche il database per verificare che l'utente sia elencato tra i moderatori. Se l'utente compare nell'elenco la flag verrà impostata a true.

2.2 Modifica delle recensioni



context UtenteStandard :: modifica(Recensione)

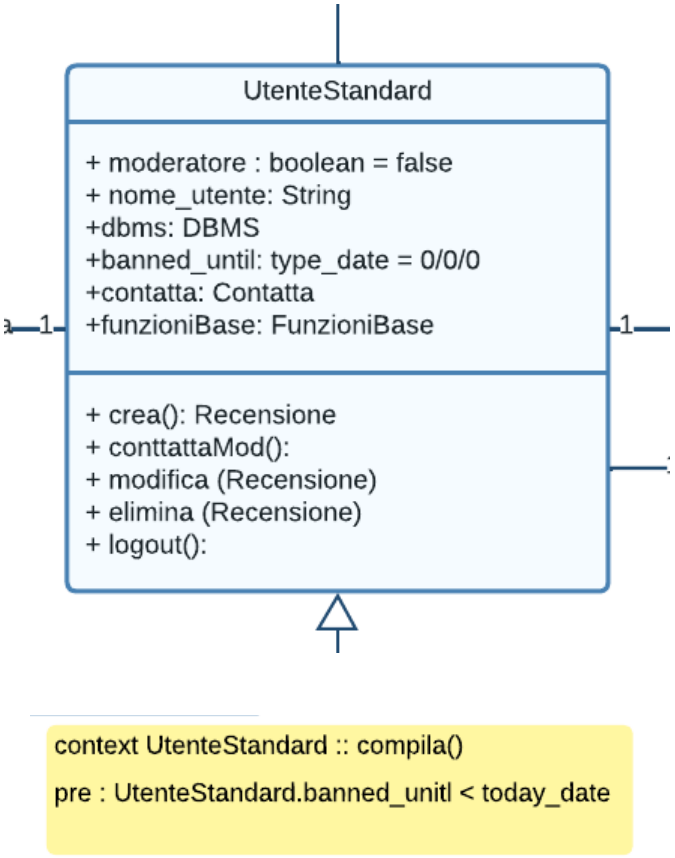
pre : (UtenteStandard.nome_utente = Recensione.autore) OR (UtenteStandard.moderatore = true)

context UtenteStandard :: elimina(Recensione)

pre : (UtenteStandard.nome_utente = Recensione.autore) OR (UtenteStandard.moderatore = true)

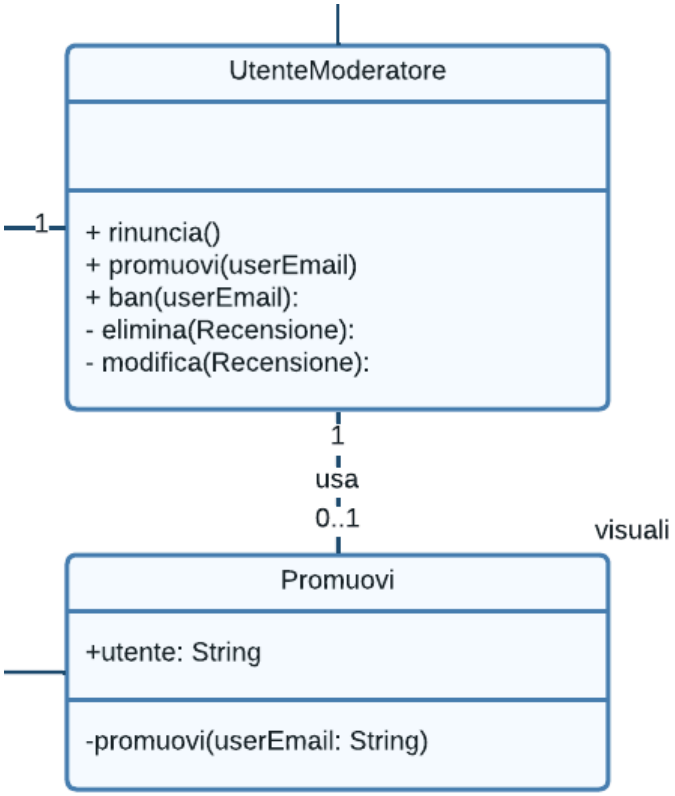
La funzione di modifica o eliminazione di una recensione è esclusiva dei soli utenti autori della recensione o agli utenti moderatori.

2.3 Compilazione della recensione



Per permettere solo agli utenti senza limitazioni di funzionalità (ban) di poter compilare recensioni, la classe UtenteStandard ha una data fino alla quale le sue funzionalità sono limitate. Se la data è tuttavia precedente al giorno in cui l'utente prova a compilare una recensione allora significa che non ha limitazioni e che può procedere.

2.4 Promozione di un utente



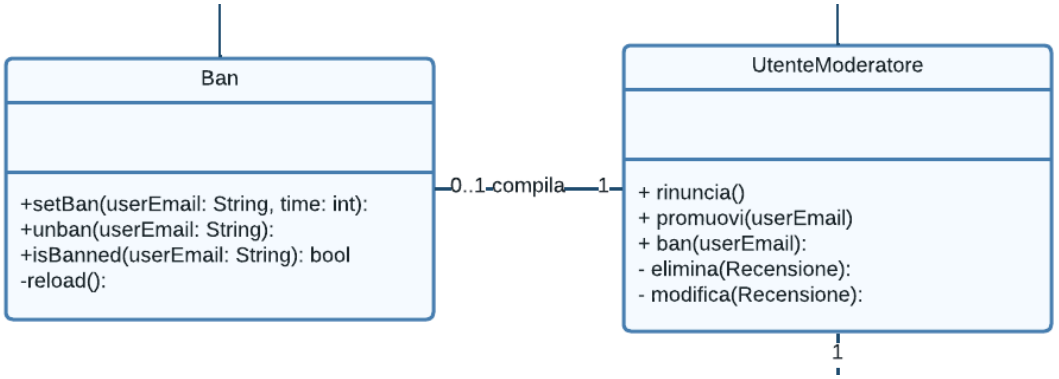
context : UtenteModeratore :: promuovi(UtenteStandard.nome_utente)

pre : UtenteStandard.moderatore = false

post : UtenteStandard.moderatore = true

In seguito alla promozione di un utente standard a moderatore, viene fatta una ricerca sul database per controllare che l'utente non sia già inserito nell'elenco degli utenti moderatori sul database. Nel caso non fosse presente questo viene aggiunto e la sua flag "moderatore" viene impostata a true.

2.5 Ban di un utente



context : UtenteModeratore :: ban(UtenteStandard.nome_utente)

pre : UtenteStandard.moderatore = false

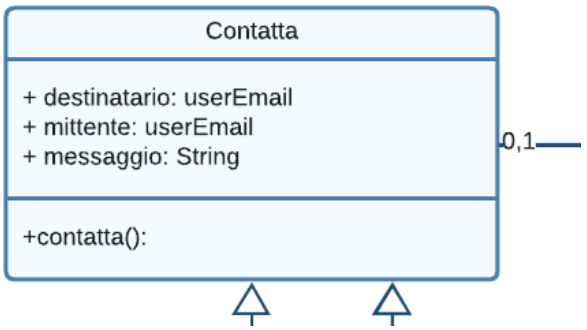
post : UtenteStandard.banned_until = new data_type

Per evitare che ai moderatori vengano limitate le funzionalità, è possibile impostare il ban solo agli utenti con flag moderatore uguale a true. Alla fine della procedura di ban viene data all'utente una nuova data fino alla quale le sue limitazioni risultano limitate.

2.6 Contatto via email

context : Contatta :: contatta()

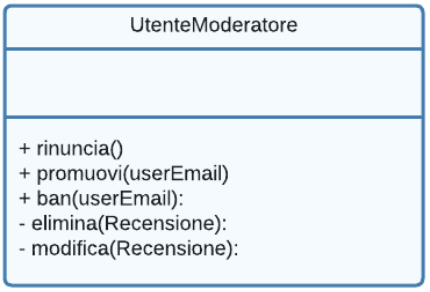
pre : Contatta.destinatario != Contatta.mittente



Non è permesso inserire lo stesso indirizzo email come destinatario e come mittente nel modulo di contatto.

2.7 Rinuncia dei privilegi da moderatore

context : UtenteModeratore :: rinuncia()
pre : UtenteStandard.moderatore = true
post : UtenteStandard.moderatore = false



Affinché un utente possa rinunciare ai suoi privilegi da moderatore, deve essere, nel momento della richiesta, moderatore.

3. DIAGRAMMA DELLE CLASSI CON OCL

Riportiamo infine il diagramma delle classi con tutte le classi fino ad ora presentate ed il codice OCL individuato.