

Documento di Architettura

Matteo Girardi, Vasile Donmovil

Gruppo T45

2023

Contents

1	Scopo del documento	3
2	Diagramma delle Classi	4
2.1	Utenti	4
2.2	Gestione Autenticazione e Account	6
2.3	Creazione e Gestione Recensioni	7
2.4	Visualizzazione Recensioni	8
2.5	Gestione Preferiti	9
2.6	Diagramma delle classi complessivo	10
3	Codice in Object Constraint Language	11
3.1	Operazioni Account Utente	11
3.2	Autorizzazioni Admin	11
3.3	Autorizzazioni Creazione Nuova Recensione	12
3.4	Autorizzazioni Gestione Recensione	12
3.5	Gestione Preferiti	12
4	Diagramma delle Classi con codice OCL	13

1 Scopo del documento

Nel presente documento viene definita l'architettura del sistema "NovelNotes" utilizzando il Class Diagram UML e codice in OCL (Object Constraint Language). Nel precedente documento sono stati rappresentati i componenti del sistema che ora verranno tradotti in classi attraverso il diagramma UML indicato.

2 Diagramma delle Classi

Nel presente capitolo vengono rappresentate le classi previste nell'ambito del progetto "NovelNotes". Ogni componente presente nel diagramma dei componenti diventa una classe. Tutte le classi individuate sono caratterizzate da un nome, una lista di attributi e una lista di metodi che definiscono le operazioni previste all'interno della classe. Associazioni tra le diverse classi forniscono informazioni sulle relazioni tra esse. Riportiamo di seguito le classi individuate.

2.1 Utenti

Analizzando il diagramma di contesto e quello dei componenti si individua la presenza di diversi attori per definire diversi livelli di "Utente".

In particolare si identificano lo user non autenticato **UNA** da cui deriva quello autenticato **UA** e i tre da esso derivati: Ordinario **UO**, Moderatore **UM** ed Admin **ADMIN**.

Utente Non Autenticato (UNA):

L'Utente Non Autenticato è un utente che visita il sito senza aver effettuato l'accesso o registrato un account. Le azioni disponibili per questo tipo di utente sono limitate. Può visualizzare le recensioni e le informazioni sui libri, ma non può interagire attivamente con il sistema, come scrivere recensioni, commentare o accedere a funzionalità riservate agli utenti autenticati.

Utente Autenticato (UA):

L'Utente Autenticato è un utente che ha effettuato l'accesso al sistema. Ha accesso a un set più ampio di funzionalità rispetto all'Utente Non Autenticato.

Utente Ordinario (UO):

L'Utente Ordinario è un tipo di Utente Autenticato che può interagire attivamente con il sistema. Ha la possibilità di scrivere recensioni sui libri, commentare le recensioni degli altri utenti e partecipare alle discussioni sulla piattaforma. Può anche cercare libri, gestire la propria lista di libri preferiti e ricevere raccomandazioni personalizzate in base alle proprie preferenze di lettura.

Utente Moderatore (UM):

L'Utente Moderatore è un tipo di Utente Autenticato con privilegi aggiuntivi. Oltre alle funzionalità dell'Utente Ordinario, ha la responsabilità di moderare il contenuto generato dagli utenti. Può prendere provvedimenti appropriati come la rimozione dei contenuti non conformi.

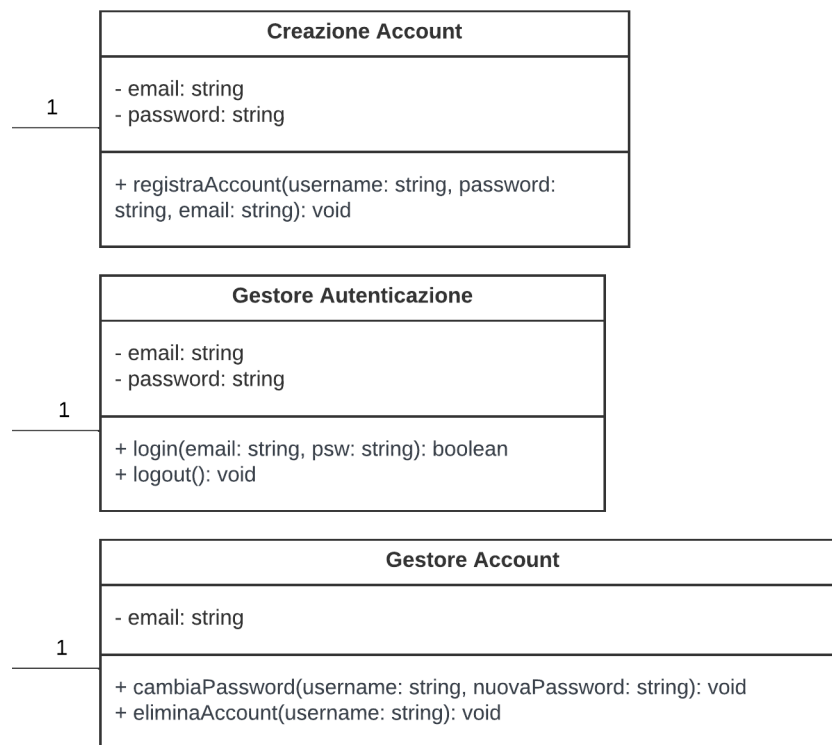
Utente Admin (ADMIN):

L'Utente Admin è il tipo di Utente Autenticato con il massimo livello di autorità nel sistema. Ha accesso completo a tutte le funzionalità e ha la possibilità di gestire gli utenti, le recensioni. Può aggiungere o rimuovere utenti, assegnare ruoli o privilegi e risolvere eventuali problemi tecnici che potrebbero sorgere.

Le diverse categorie di utenti nel sistema di recensioni di libri hanno ruoli e responsabilità distinte che consentono loro di partecipare attivamente, gestire e mantenere un ambiente di discussione e condivisione di contenuti costruttivo e di qualità.

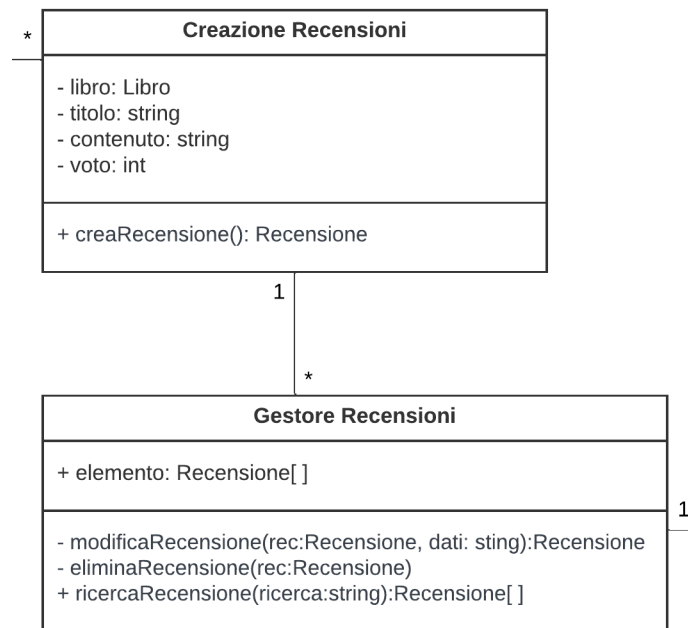
2.2 Gestione Autenticazione e Account

Il diagramma dei componenti include diversi componenti per la gestione degli account a partire dalla creazione di un nuovo account, passando dall'autenticazione e permettendo la gestione del proprio account. Definiamo quindi le classi CreazioneAccount, GestoreAutenticazione e GestoreAccount. La classe CreazioneAccount viene utilizzato unicamente dall'utente non ancora registrato, permettendo poi all'utente di autenticarsi. Il GestoreAccount presenta diversi metodi usati sia da utenti normali che da Admin e Moderatore con l'autorizzazione specifica.



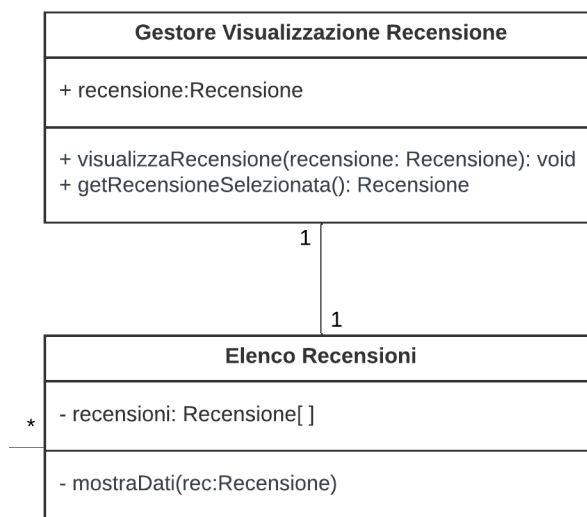
2.3 Creazione e Gestione Recensioni

Il diagramma dei componenti identifica due componenti per la gestione delle recensioni, dividendo il processo in una prima parte di creazione e una seconda di gestione. Avendo accesso all'intero elenco di recensioni la gestione si occupa anche di effettuare la ricerca di una recensione. Vengono quindi identificate le rispettive classi Creazione Recensione e Gestore Recensioni.



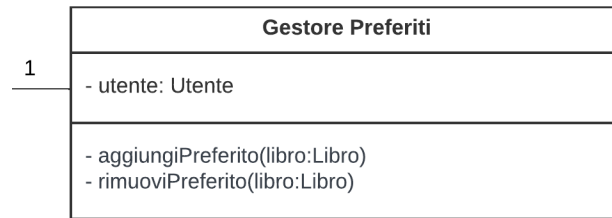
2.4 Visualizzazione Recensioni

Il diagramma dei componenti identifica i componenti Gestore Visualizzazione Recensioni e Elenco Recensioni che assieme si occupano di mostrare le recensioni. Creando una classe per ogni componente identifichiamo Elenco Recensioni e GestoreVisualizzazioneRecensione. L'utente si interfaccia con queste classi per ricercare e visualizzare le recensioni. Può visualizzare i dati della singola recensione mostrati dal GestoreVisualizzazioneRecensioni.



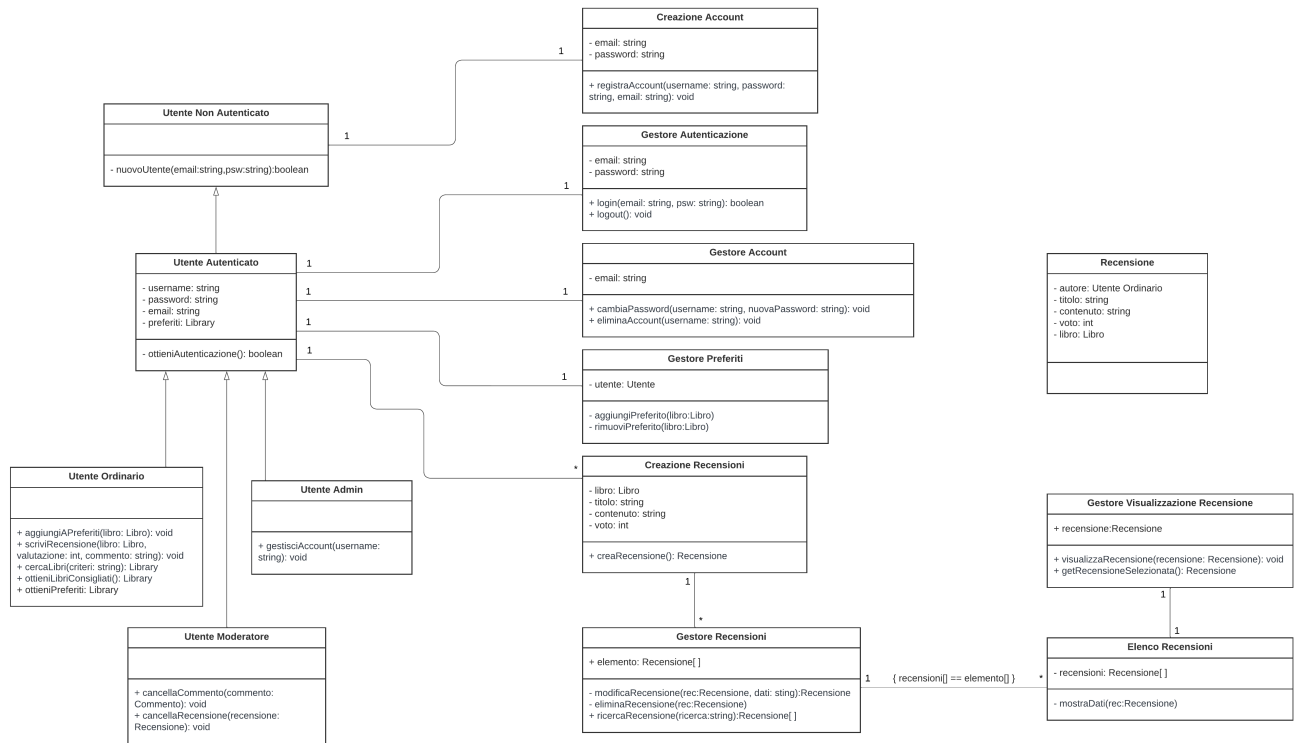
2.5 Gestione Preferiti

Il diagramma dei componenti identifica un gestore dei preferiti per permettere all'utente autenticato di salvarsi i libri a cui è interessato. Identifichiamo quindi la classe `GestorePreferiti`.



2.6 Diagramma delle classi complessivo

Riportiamo di seguito il diagramma delle classi con tutte le classi presentate. È presente inoltre una classe ausiliaria, "Recensione", per descrivere il tipo di dato Recensione utilizzato da altre classi nel diagramma.



3 Codice in Object Constraint Language

In questo capitolo è descritta in modo formale la logica prevista nell'ambito di alcune operazioni di alcune classi. Tale logica viene descritta in Object Constraint Language (OCL).

3.1 Operazioni Account Utente

È presente una serie di restrizioni in base a se l'utente ha già effettuato il login oppure no e in base al tipo di utente. In particolare non è possibile creare un utente se ne esiste già uno con la stessa mail; la modifica della password, l'eliminazione dell'account e il logout richiedono di aver già effettuato l'accesso; Il logout cambia lo stato da autenticato a non autenticato; il reset della password e il login richiedono che l'utente non sia già autenticato (e il login cambia lo stato da non autenticato ad autenticato).

```
context CreazioneAccount.registraAccount()
  pre UtenteAutenticato.email = null
context GestoreAccount::cambiaPassword()
  pre UtenteAutenticato::ottieniAutenticazione()=1
context GestoreAccount::eliminaAccount()
  pre UtenteAutenticato::ottieniAutenticazione()=1
  post UtenteRegistrato = null
context GestoreAutenticazione::login()
  pre UtenteAutenticato::ottieniAutenticazione()=0
  post UtenteAutenticato::ottieniAutenticazione()=1
context GestoreAutenticazione::logout()
  pre UtenteAutenticato::ottieniAutenticazione()=1
  post UtenteAutenticato::ottieniAutenticazione()=0
```

3.2 Autorizzazioni Admin

Solo l'admin ha accesso alla consegna del ruolo di Moderatore agli account e alla revoca degli stessi, previa sua autenticazione.

```
context GestoreAccount::promuoviMod()
  pre UtenteAutenticato::ottieniAutenticazione()=1
  pre UtenteAdmin.isAdmin=1
context GestoreAccount::declassaMod()
  pre UtenteAutenticato::ottieniAutenticazione()=1
  pre UtenteAdmin.isAdmin=1
```

3.3 Autorizzazioni Creazione Nuova Recensione

Per poter creare una nuova recensione è necessario che l'utente sia autenticato.

```
context CreazioneRecensioni::creaRecensione()
  pre UtenteAutenticato::ottieniAutenticazione()=1
```

3.4 Autorizzazioni Gestione Recensione

Per poter modificare o eliminare una recensione è necessario essere moderatore oppure essere l'utente proprietario della recensione che si ha intenzione di eliminare/modificare.

```
context GestoreRecensioni::modificaRecensione()
  pre UtenteAutenticato::ottieniAutenticazione()=1
  pre UtenteAutenticato::isModerator()=1 OR Recensione.autore = UtenteAutenticato.username
context GestoreEventi::eliminaRecensione()
  pre UtenteAutenticato::ottieniAutenticazione()=1
  pre UtenteAutenticato::isModerator()=1 OR Recensione.autore = UtenteAutenticato.username
```

3.5 Gestione Preferiti

Il gestore preferiti permette unicamente all'utente autenticato di aggiungere un libro alla lista dei libri preferiti se questo non è già presente e di rimuoverlo in caso contrario.

```
context GestorePreferiti::aggiungiPreferito()
  pre UtenteAutenticato::ottieniAutenticazione()=1
  pre Libro not in UtenteAutenticato.preferiti[]
  post Libro in UtenteAutenticato.preferiti[]
context GestorePreferiti::rimuoviPreferito()
  pre UtenteAutenticato::ottieniAutenticazione()=1
  pre Libro in UtenteAutenticato.preferiti[]
  post Libro not in UtenteAutenticato.preferiti[]
```

4 Diagramma delle Classi con codice OCL

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

