

Comic Reader

Francesco Panattoni

15 Luglio 2024

Progetto di Programmazione Avanzata
A.A 23-24

1	Introduzione	1
2	Lato Server	2
2.1	Struttura del Database	2
2.2	Interazione col Database	3
2.3	API	4
2.4	Test	4
2.5	Gestore Fumetti	5
2.5.1	Primo Avvio del Server	5
2.5.2	Inserimento di un Fumetto in Locale	5
3	Lato Client	9
3.1	Login e Signup	9
3.1.1	Dettagli Implementativi	11
3.2	Pagina Principale	12
3.2.1	Dettagli Implementativi	13
3.3	Lettura dei Fumetti	14
3.3.1	Dettagli Implementativi	15
4	Conclusione	15



UNIVERSITÀ DI PISA



COMIC READER

LEGGI I TUOI FUMETTI PREFERITI!

1 Introduzione

Comic Reader è un'applicazione sviluppata per facilitare la lettura dei vostri fumetti preferiti, caratterizzata da un'interfaccia intuitiva e user-friendly. L'applicazione offre diverse funzionalità per migliorare l'esperienza di lettura dell'utente.

Oltre a potere leggere fumetti e manga con facilità, Comic Reader tiene traccia di ciò che leggi e in particolare memorizza l'ultimo capitolo letto, aiutandoti qualora non ricordassi dove sei rimasto nella lettura. Inoltre, permette di creare una lista di fumetti che si desidera leggere in futuro all'interno dell'applicazione, rapida da consultare e da gestire. È anche possibile dare una valutazione personale al fumetto letto, da 1 a 10.

Un'altra funzionalità di rilievo è la possibilità di leggere manga utilizzando la modalità di “lettura orientale” (da destra verso sinistra), offrendo così un'esperienza autentica per gli appassionati del genere.

L'applicazione include un fumetto predefinito, inserito tramite una richiesta GET ad un'API¹ e offre la possibilità di aggiungere fumetti localmente, formando così una propria collezione digitale.

Nel prosieguo della presentazione, esamineremo la struttura dell'applicazione, approfondendo non solo tutte le sue funzionalità, ma anche alcuni dettagli tecnici e implementativi.

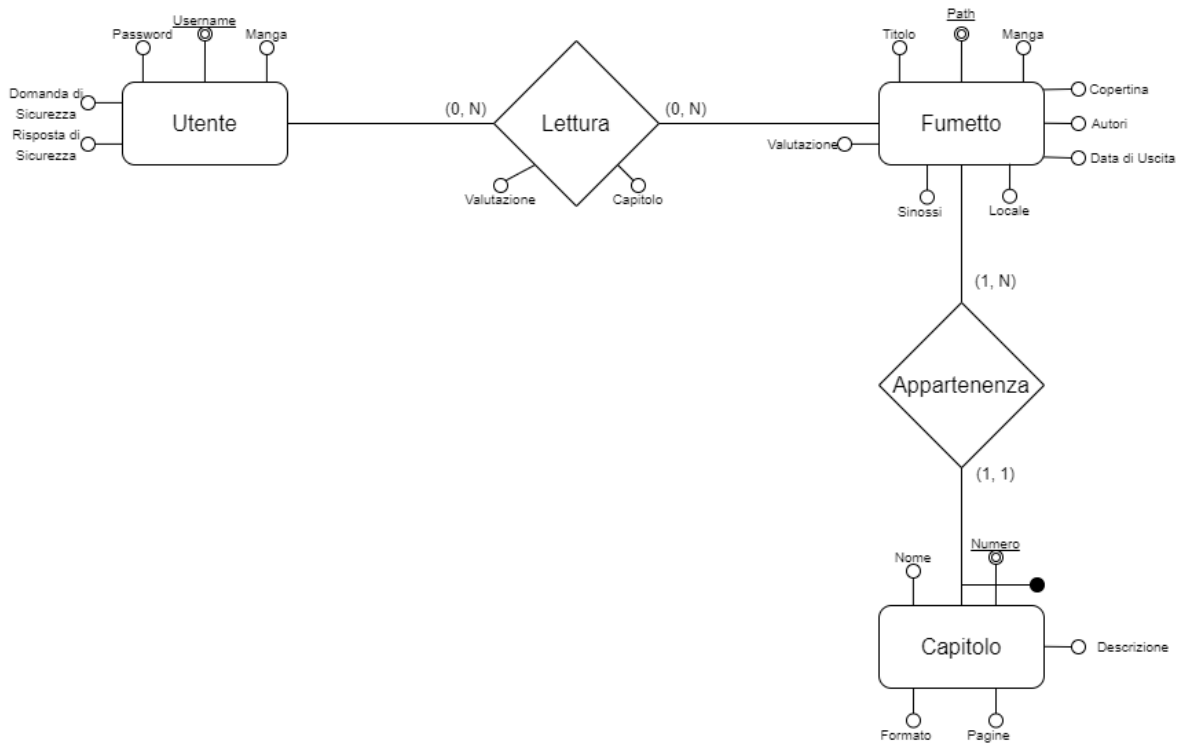
¹ <https://xkcd.com/info.0.json>

2 Lato Server

Passiamo ora alla discussione della componente server dell'applicazione.

2.1 Struttura del Database

Questa è la raffigurazione del Diagramma E-R del Database d604230:



Il database è formato da 4 tabelle:

- **Utente:** contiene i dati degli utenti che hanno effettuato la registrazione.
 - L'attributo *manga* è un booleano che indica se la lettura orientale è attivata.
- **Lettura:** relazione molti a molti tra Utente e Fumetto, diventa una tabella che contiene i fumetti letti dall'utente o i fumetti che l'utente vuole leggere.
 - L'attributo *capitolo* è un intero maggiore o uguale a 0. Se messo a 0, significa che l'utente desidera leggere questo fumetto in futuro. Altrimenti se maggiore di 0, è il capitolo dove l'utente è arrivato a leggere.
 - L'attributo *valutazione* è un intero compreso tra 1 e 10 e indica la valutazione data dall'utente al fumetto.
- **Fumetto:** contiene i dati relativi al fumetto, tra cui *Path* che è il percorso all'interno della cartella comic-reader-server che contiene i capitoli del fumetto.
 - L'attributo *copertina* indica il percorso all'interno della cartella comic-reader-server che contiene la copertina del fumetto da mostrare all'utente.
 - L'attributo *manga* è un booleano che indica se il fumetto è un manga o meno. Se è un manga può essere letto tramite la lettura orientale.
 - L'attributo *valutazione* è un intero che indica la valutazione complessiva degli utenti che usufruiscono di questa applicazione.
 - L'attributo *locale* indica se il fumetto è stato inserito localmente o è stato scaricato tramite API.
- **Capitolo:** contiene i dati relativi ad un capitolo di un fumetto e tiene in considerazione quante pagine ha e di che formato sono le pagine di cui è composto (.jpg, .png o .jpeg). È una cartella con dentro delle immagini.
 - La relazione Appartenenza viene accorpata a Capitolo in modo da avere *Path* nella chiave insieme al *Numero* del capitolo.

2.2 Interazione col Database

La modellazione e l'interazione con il database sono state realizzate utilizzando Spring JPA, creando un Java Bean per ogni tabella. Sebbene siano stati impiegati alcuni metodi nativi di Spring JPA per le query, per funzionalità più specifiche sono state implementate query native.

Come da specifiche del progetto le credenziali di accesso a MySQL sono:

- IP: 127.0.0.1 (localhost)
- User: root
- Password: root

Per la creazione del database all'avvio, in caso di inesistenza, è stata utilizzata una funzionalità Spring JPA di interfaccia con MySQL:

```
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/d604230?createDatabaseIfNotExist=true
```

Metodi Nativi Utilizzati:

- Utente:
 - findByUsername(String user);
- Lettura:
 - findById_UtenteAndID_Fumetto(Utente utente, Fumetto fumetto);
- Fumetto:
 - findByPath(String path);
 - findByCopertina(String copertina);
- Capitolo:
 - findById(CapitoloID ID);

Query Native Utilizzate:

- Lettura:
 - @Query("SELECT AVG(l.valutazione) FROM Lettura l WHERE l.ID.fumetto = :fumetto AND l.valutazione IS NOT NULL");
 - @Query("SELECT DISTINCT l.ID.fumetto FROM Lettura l WHERE l.ID.utente = :utente");
- Fumetto:
 - @Query(value = "SELECT COUNT(*) AS Capitoli FROM Fumetto F INNER JOIN Capitolo C ON F.Path = C.Fumetto WHERE F.Path = :path", nativeQuery = true);
 - @Query(value = "SELECT * FROM Fumetto F WHERE F.titolo LIKE %:testo% OR F.autori LIKE %:testo% LIMIT 15", nativeQuery = true);
 - @Query(value = "SELECT * FROM Fumetto F WHERE F.titolo LIKE %:testo% AND F.Locale IS TRUE LIMIT 15", nativeQuery = true);
- Capitolo:
 - @Query(value = "SELECT * FROM Capitolo C WHERE C.Fumetto LIKE %:p% ORDER BY C.Numero", nativeQuery = true);
 - @Query(value = "SELECT COUNT(*) AS Capitoli FROM Capitolo C WHERE C.Fumetto = :p", nativeQuery = true);

Sia la password, che la risposta di sicurezza vengono criptate attraverso *org.mindrot.jbcrypt.BCrypt*.

2.3 API

Il Server di Comic Reader offre le seguenti API accessibili tramite protocollo HTTP, con lo scambio di dati codificati in formato JSON utilizzando `CodiceRisposta.java` o `ResponseEntity<>`². Viene usata la porta 8080. Le specifiche dettagliate sono disponibili nel Javadoc.

Utente

- (POST) /utente/login
- (POST) /utente/signup
- (GET) /utente/forgot
- (POST) /utente/cambioPwd
- (POST) /utente/cambioUsr
- (POST) /utente/cambioSec
- (POST) /utente/mangaLet
- (GET) /utente/getManga

Lettura

- (POST) /lettura/add
- (GET) /lettura/rating
- (POST) /lettura/update
- (POST) /lettura/remove
- (GET) /lettura/search

Fumetto

- (POST) /fumetto/add
- (GET) /fumetto/search
- (GET) /fumetto/local
- (GET) /fumetto/recover
- (GET) /fumetto/getManga
- (POST) /fumetto/update
- (POST) /fumetto/cover

Capitolo

- (POST) /capitolo/add
- (GET) /capitolo/recover
- (POST) /capitolo/read
- (GET) /capitolo/chapters
- (GET) /capitolo/pages
- (POST) /capitolo/removeComic
- (POST) /capitolo/update

2.4 Test

I test sono implementati nel file `UtenteControllerTest.java`, dove si controlla il corretto funzionamento del login, della registrazione, del cambio dell'username, del cambio di password, del cambio delle credenziali e per il cambio dell'attributo *manga* per la lettura orientale.

² `org.springframework.http.ResponseEntity`

2.5 Gestore Fumetti

La classe Java *GestoreFumetti* è progettata per gestire l'inserimento e la gestione di fumetti e capitoli utilizzando il Singleton Pattern. Viene avviato automaticamente subito dopo l'avvio del server.

La classe *GestoreFumetti* integra diverse funzionalità per gestire fumetti e capitoli sia online (utilizzando API esterne per recuperare e memorizzare informazioni sui fumetti), che localmente. Implementa un approccio asincrono per l'inserimento dei fumetti e dei capitoli tramite thread separati. L'inserimento dei dati nel database avviene attraverso l'API precedentemente fornite.

Questa classe scarica le pagine di un fumetto dal sito XKCD, inclusa l'immagine della copertina e i capitoli (associati agli anni di pubblicazione). Utilizza le classi *Fumetto* e *Capitolo* per rappresentare le cartelle create per ospitare il fumetto XKCD e i capitoli, con dentro le immagini che a loro volta rappresentano le pagine del fumetto.

Inoltre gestisce anche l'inserimento dei fumetti locali. Per aggiungere un nuovo fumetto all'applicazione, è necessario posizionare una cartella all'interno di "comic-reader-server\src\main\fumetti", la quale deve contenere sottocartelle rappresentanti i capitoli (già ordinate per titolo in modo che la prima cartella rappresenti il primo capitolo, la seconda il secondo e così via). Ogni sottocartella dei capitoli deve contenere immagini ordinate, dove la prima immagine corrisponde alla prima pagina, la seconda immagine alla seconda pagina e così via. È possibile modificare la copertina del fumetto inserendo un'immagine nella cartella "comic-reader-server\src\main\fumetti\Copertine" e poi richiedendo un cambiamento della copertina del fumetto tramite l'applicazione (tutto ciò è approfondito nella sezione 2.5.2).

Inoltre il Gestore Fumetti si occupa di vedere se ci sono capitoli nuovi per aggiungerli automaticamente al database.

2.5.1 Primo Avvio del Server

Al primo avvio del server, verrà scaricato il fumetto XKCD. Per ridurre il tempo di download, non verranno scaricate tutte le immagini presenti sul sito, ma solo alcune, suddivise per anno (dal 2022 al 2024). Il processo di download delle immagini richiederà un tempo compreso tra 10 e 30 secondi.

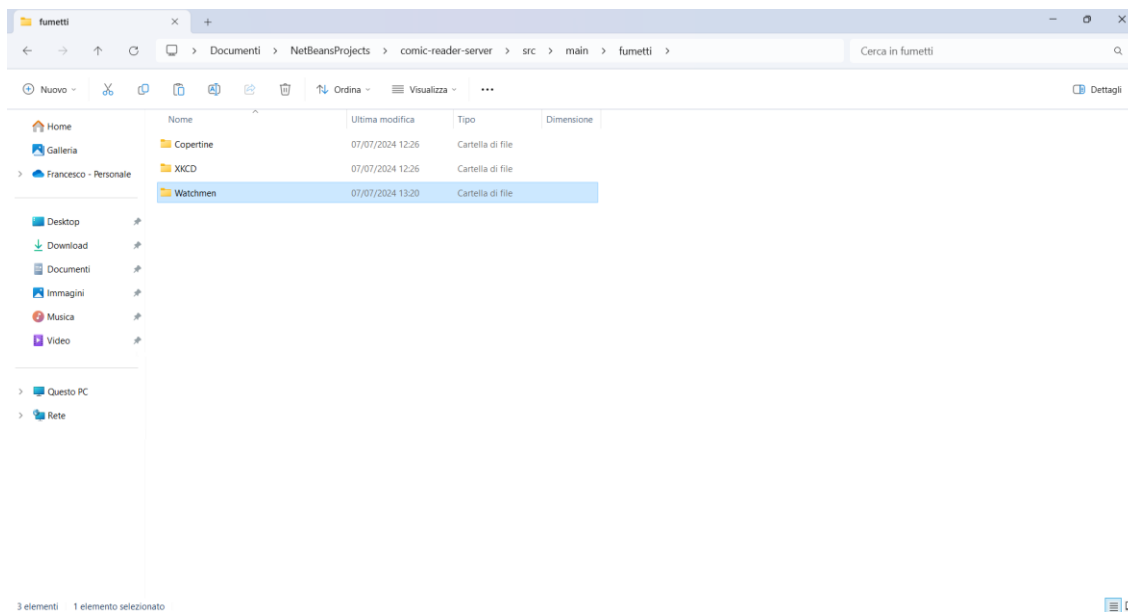
Una volta completato il download, verrà creata la cartella denominata "fumetti" con all'interno le sottocartelle "Copertine" e "XKCD". La cartella "XKCD" conterrà ulteriori tre sottocartelle: "1", "2" e "3". Esse rappresentano i tre capitoli disponibili del fumetto XKCD. Nel Database saranno memorizzati correttamente i dati e la cartella "1" rappresenterà le immagini dell'anno 2022, la cartella "2" dell'anno 2023 e la cartella "3" dell'anno 2024. Dentro queste cartelle ci sono le immagini, ossia le pagine del fumetto, numerate partendo da 1 e con estensione .png.

La copertina del fumetto XKCD sarà inserita nella cartella "Copertine" e denominata con il nome del fumetto stesso ("XKCD.jpg"). In aggiunta, sarà presente un'altra immagine denominata "Comic.jpg", utilizzata come copertina predefinita per i fumetti che non hanno una copertina assegnata. XKCD è considerato un manga solo per mostrare il funzionamento della lettura orientale.

2.5.2 Inserimento di un Fumetto in Locale

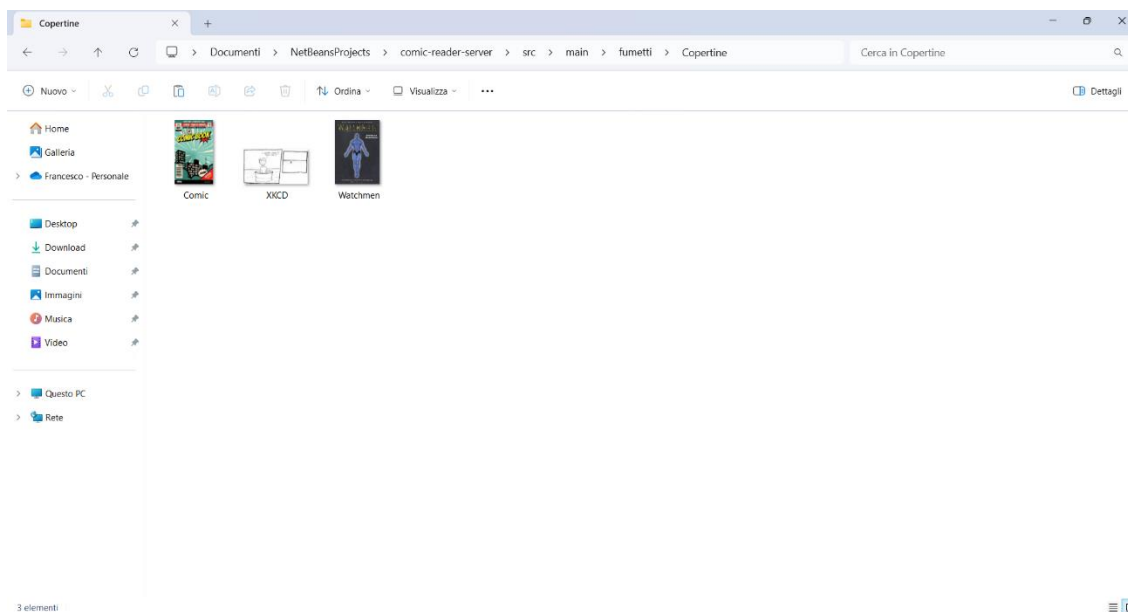
Quando la componente server non è in funzione, è possibile aggiungere un fumetto. È necessario accedere al percorso: "comic-reader-server\src\main\fumetti". Una volta raggiunto, bisogna copiare la cartella del fumetto all'interno. Il nome della cartella verrà utilizzato come titolo del fumetto.

All'interno della cartella del fumetto, devono essere presenti le sottocartelle, organizzate in capitoli e ordinate in ordine crescente. Le immagini del fumetto devono essere collocate nelle rispettive cartelle dei capitoli, già ordinate e con lo stesso formato. Il Gestore Fumetti si occuperà di rinominare le immagini in modo appropriato e di inserire i capitoli e il fumetto nel database.



Una volta fatto ciò, se il procedimento è stato eseguito correttamente, sarà possibile inserire il fumetto nell'applicazione. Avviamo la componente server e il fumetto è ora disponibile da leggere. Una volta fatto il login sarà disponibile, ma con la copertina di default, senza l'autore, la data di uscita e la sinossi.

Per la copertina dobbiamo inserirla nella cartella Copertine. Copiamoci l'immagine che vogliamo come copertina all'interno.



Dopo di che serve fare il login (la registrazione se ancora non abbiamo un account, vedi il capitolo 3, sezione 3.1), cliccare sull'icona del menù a tendina in alto a sinistra e cliccare su "Locale". Lì sarà possibile cambiare le informazioni sul fumetto Watchmen.

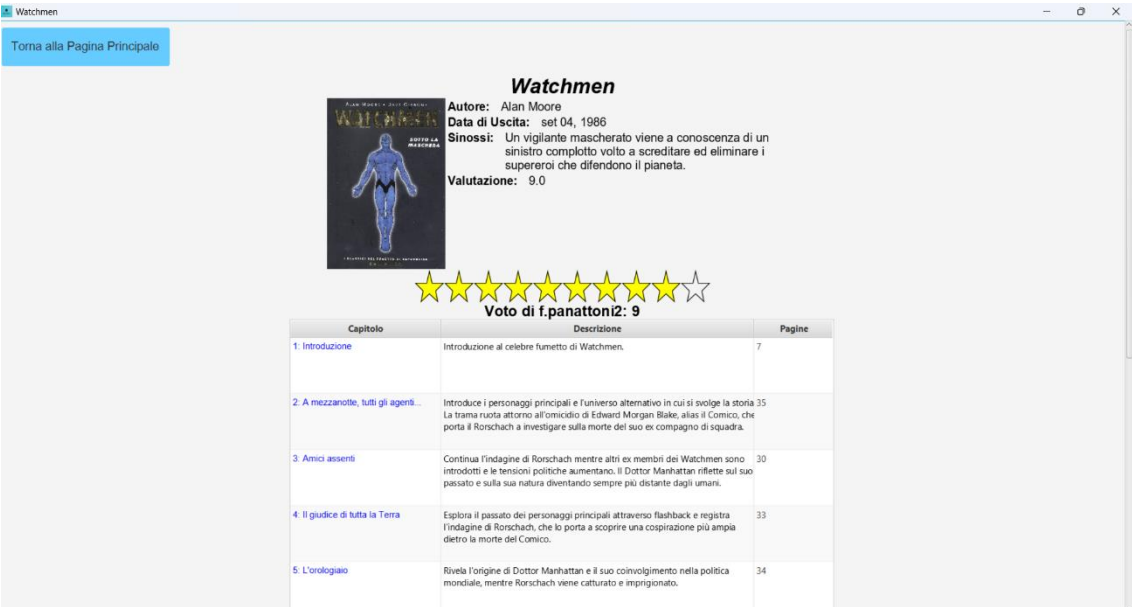
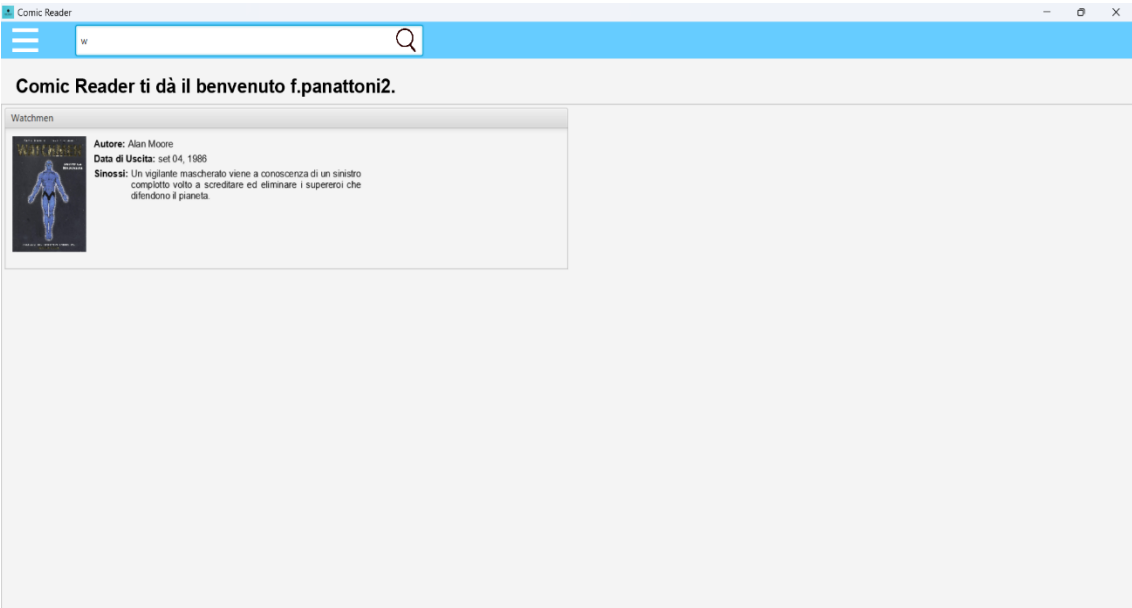


Ora è possibile inserire le informazioni riguardo al fumetto e ai suoi capitoli. Prima di cliccare “Salva”, cliccare sempre INVIO per salvare l’informazione scritta e subito dopo “Salva”. In Copertina cambiare l’ultima parte e sostituire “Comic.jpg” con l’immagine scelta.



La Data di Uscita può avere due formati: “MMM dd, yyyy” (ad esempio set 04, 1986) o “yyyy-MM-dd” (ad esempio 1986-09-04). Poi verrà sempre mostrata col formato “MMM dd, yyyy”.

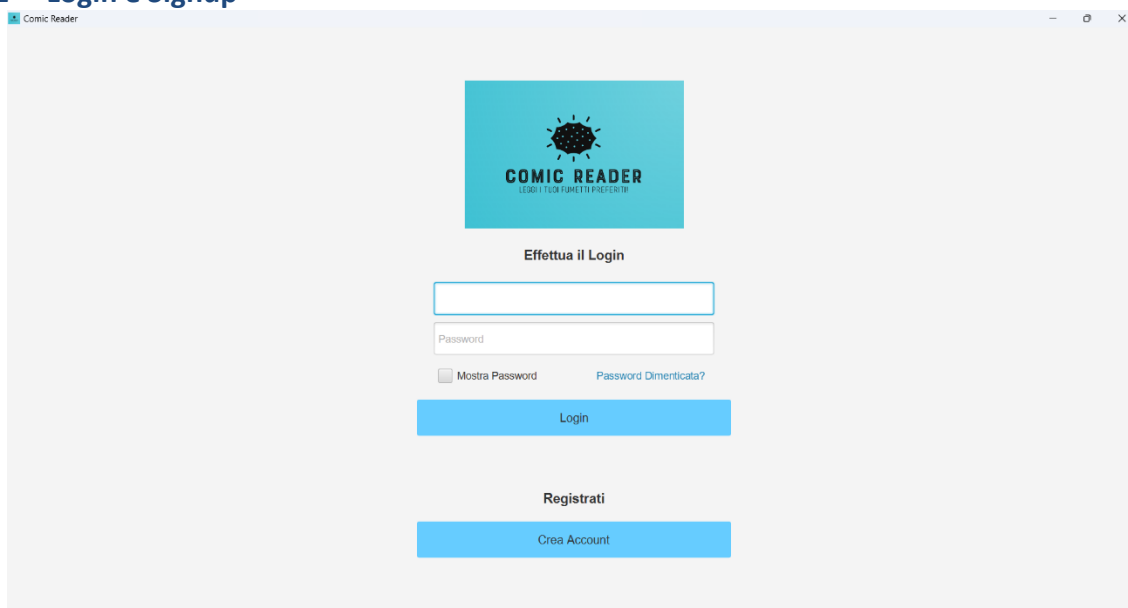
Il pulsante “Rimuovi” elimina il fumetto dal database, a condizione che non sia stato letto da altri account o che non sia stato inserito nella lista dei fumetti da leggere di altri account. In quest'ultimo caso, il fumetto viene rimosso soltanto dalla lista delle letture dell'account che ha richiesto la rimozione. È importante notare che la rimozione riguarda esclusivamente il database. In questo modo, l'utente ha la possibilità di eliminare manualmente il fumetto dalla cartella “fumetti”, o di spostarlo nel caso in cui non ne possieda un'altra copia. Bisogna anche eliminare la copertina nella cartella “Copertine”.



3 Lato Client

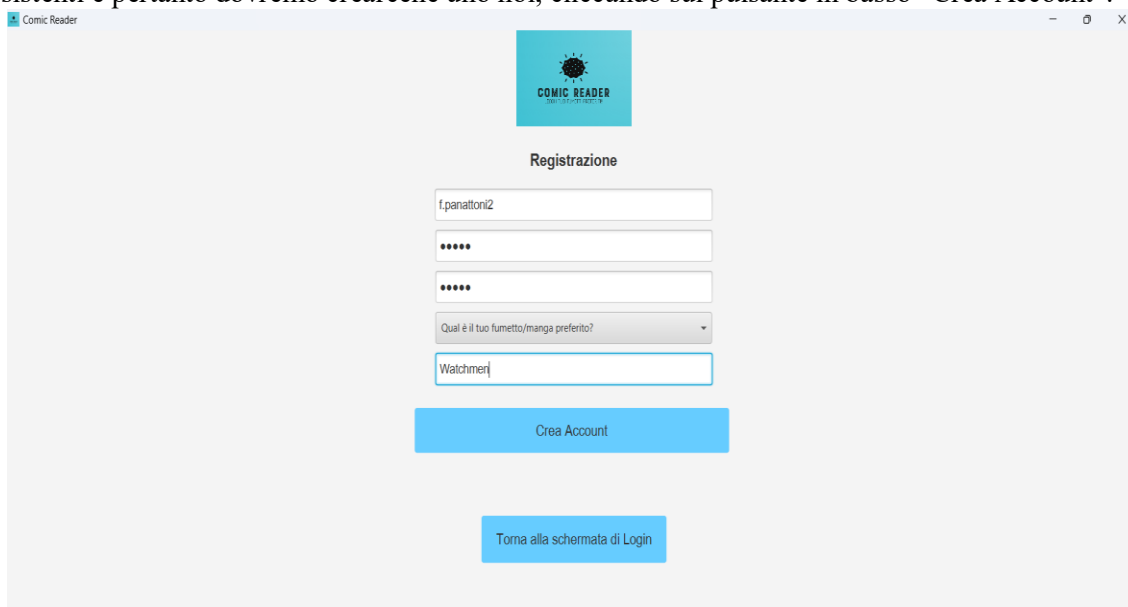
In questa sezione esamineremo anzitutto le diverse interfacce che costituiscono l'applicativo e le relative funzionalità. Una volta avviata la componente server, si può avviare la componente client. L'applicazione comunica con il servizio inviando richieste HTTP. A tal fine, viene utilizzata la classe *java.net.HttpURLConnection*. Alcune richieste richiedono l'invio di dati nel corpo della richiesta e in tali casi, i dati vengono serializzati in formato JSON.

3.1 Login e Signup



The screenshot shows the 'Comic Reader' login interface. At the top, there is a logo with a sun-like icon and the text 'COMIC READER' and 'LEGGI I TUI FUMETTI PREFERITI'. Below the logo, the text 'Effettua il Login' is displayed. There are two input fields: one for the username (containing 'f.panattoni2') and one for the password (containing '*****'). Below the password field, there is a checkbox labeled 'Mostra Password' and a link 'Password Dimenticata?'. A blue 'Login' button is positioned below the input fields. Below the 'Login' button, the text 'Registrati' is displayed, followed by a blue 'Crea Account' button.

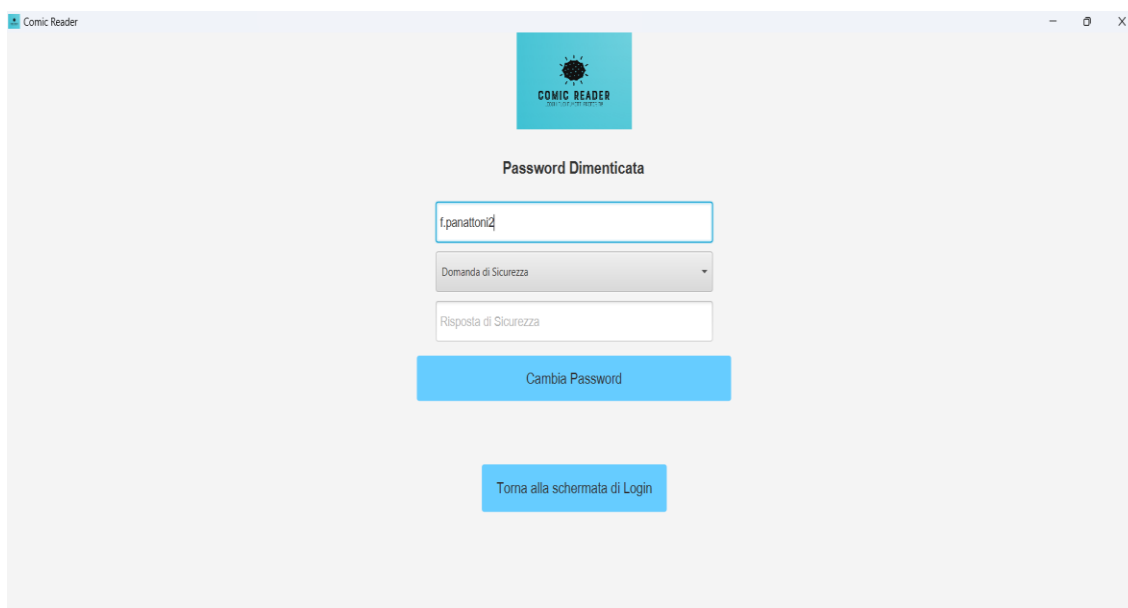
Una volta fatto l'accesso all'applicazione, ci troveremo ad una schermata di Login. Non ci sono account preesistenti e pertanto dovremo crearne uno noi, cliccando sul pulsante in basso "Crea Account".



The screenshot shows the 'Comic Reader' registration interface. At the top, there is a logo with a sun-like icon and the text 'COMIC READER' and 'LEGGI I TUI FUMETTI PREFERITI'. Below the logo, the text 'Registrazione' is displayed. There are four input fields: one for the username (containing 'f.panattoni2'), one for the password (containing '*****'), one for the confirm password (containing '*****'), and one for the favorite comic/manga (containing 'Watchmen'). Below the input fields, there is a dropdown menu labeled 'Qual è il tuo fumetto/manga preferito?'. A blue 'Crea Account' button is positioned below the input fields. Below the 'Crea Account' button, there is a blue button labeled 'Torna alla schermata di Login'.

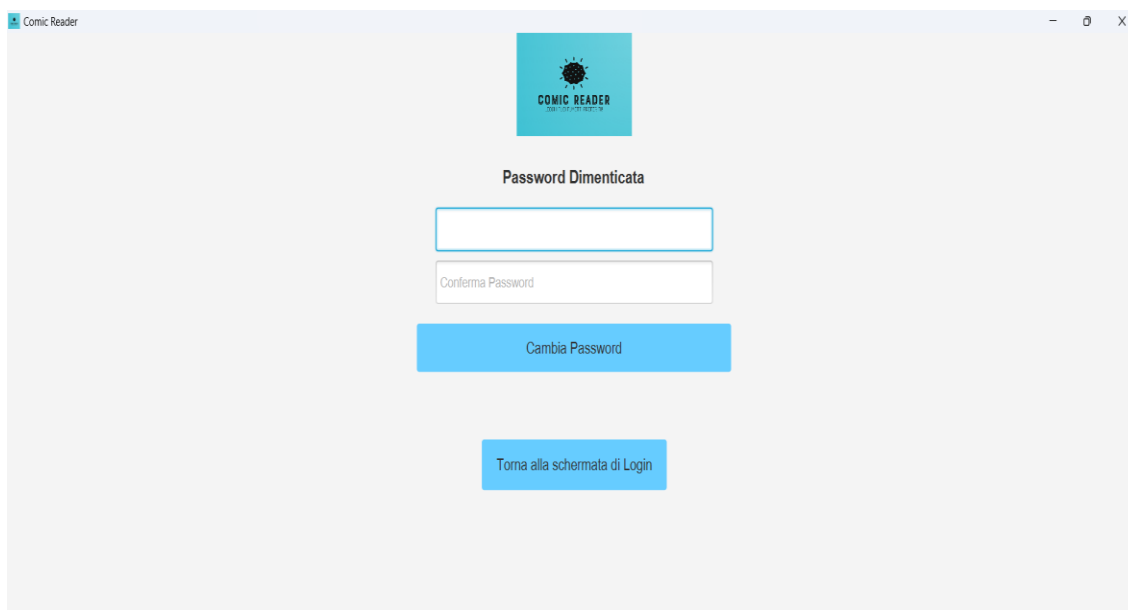
La password deve essere di almeno cinque caratteri e non deve averne più di quindici. È obbligatorio compilare tutti e cinque i campi (username, password, conferma della password, domanda di sicurezza e risposta di sicurezza). Dopo aver cliccato sul pulsante "Crea Account" saremo indirizzati verso la Pagina Principale, collegati al nostro account appena creato. Descriveremo successivamente il funzionamento della Pagina Principale. Intanto mostriamo le altre interfacce dedicate al Login.

Ritornando alla schermata di login e cliccando su "Password Dimenticata" (la scritta verrà sottolineata al passaggio del cursore), si accederà alla schermata successiva.



The screenshot shows a web browser window titled "Comic Reader". The page has a light gray background. At the top center is a logo for "COMIC READER" with a stylized sun icon. Below the logo, the title "Password Dimenticata" is centered. There are three input fields: the first is for the username, containing the text "t.panattoni"; the second is a dropdown menu labeled "Domanda di Sicurezza"; and the third is for the security question answer, labeled "Risposta di Sicurezza". Below these fields is a large blue button labeled "Cambia Password". At the bottom center is a smaller blue button labeled "Torna alla schermata di Login".

È necessario inserire l'username, la domanda di sicurezza e la risposta di sicurezza prima di poter cambiare la password. Una volta inseriti i dati corretti e cliccato sulla pulsante "Cambia Password" si passa alla schermata di Cambio Password, dove è possibile cambiarla. La password non può essere la stessa di prima.



The screenshot shows the same web browser window titled "Comic Reader". The page layout is similar to the previous one, with the "COMIC READER" logo at the top. The title "Password Dimenticata" is still present. However, the input fields are now: an empty password field and a "Conferma Password" (Confirm Password) field. The blue "Cambia Password" button is still present. At the bottom center is the same blue button labeled "Torna alla schermata di Login".

3.1.1 Dettagli Implementativi

Il package che si occupa di questa parte è “it.unipi.panattoni.client.login”.

Di questo package fanno parte:

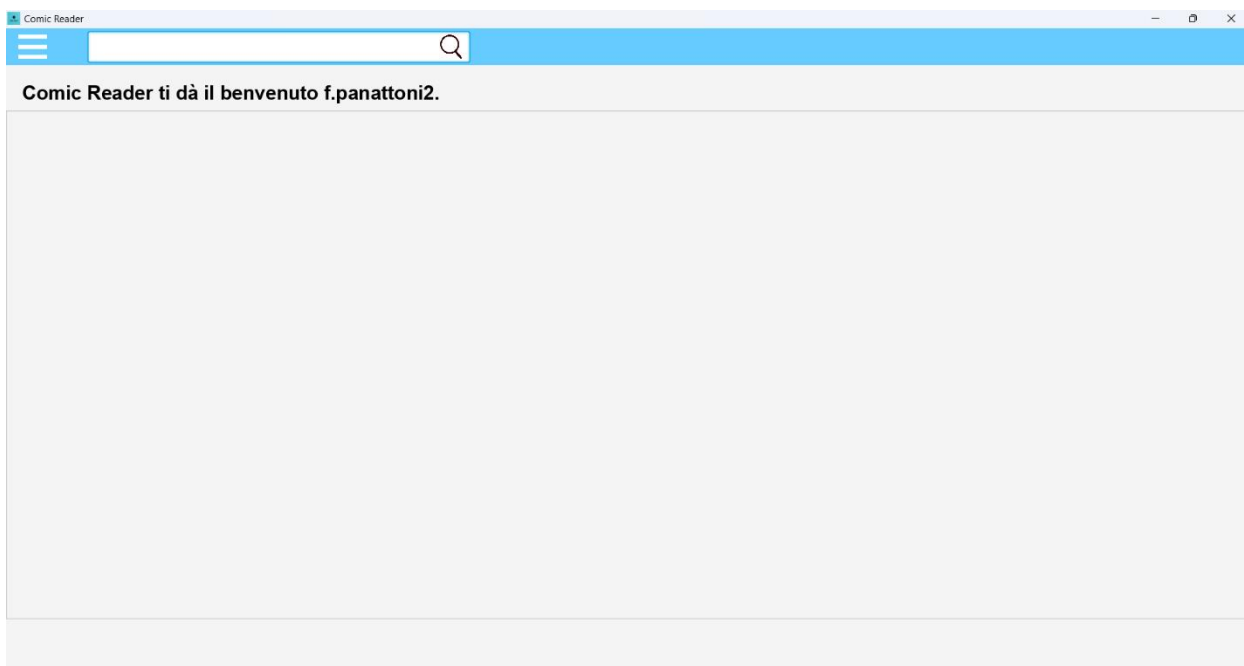
- Utente.java: Rappresenta i dati dell'Utente nel sistema;
- LoginController.java: Il controllore di login.fxml che si occupa del login dell'Utente;
- RegisterController.java: Il controllore di register.fxml che si occupa della registrazione dell'Utente;
- ForgotController.java: Il controllore di forgot.fxml che si occupa dell'accesso al cambio password dell'Utente;
- ChangePasswordController.java: Il controllore chagepwd.fxml (quello del login) che si occupa del cambio della Password dell'Utente.

Una cosa particolare è che per passare da forgot.fxml a changepwd.fxml, c'è bisogno di passare lo username come parametro (stringa³) alla classe *ChangePasswordController*. Pertanto *ChangePasswordController* non è messo come controllore di default di changepwd.fxml, ma di questo si occupa *ForgotController*.

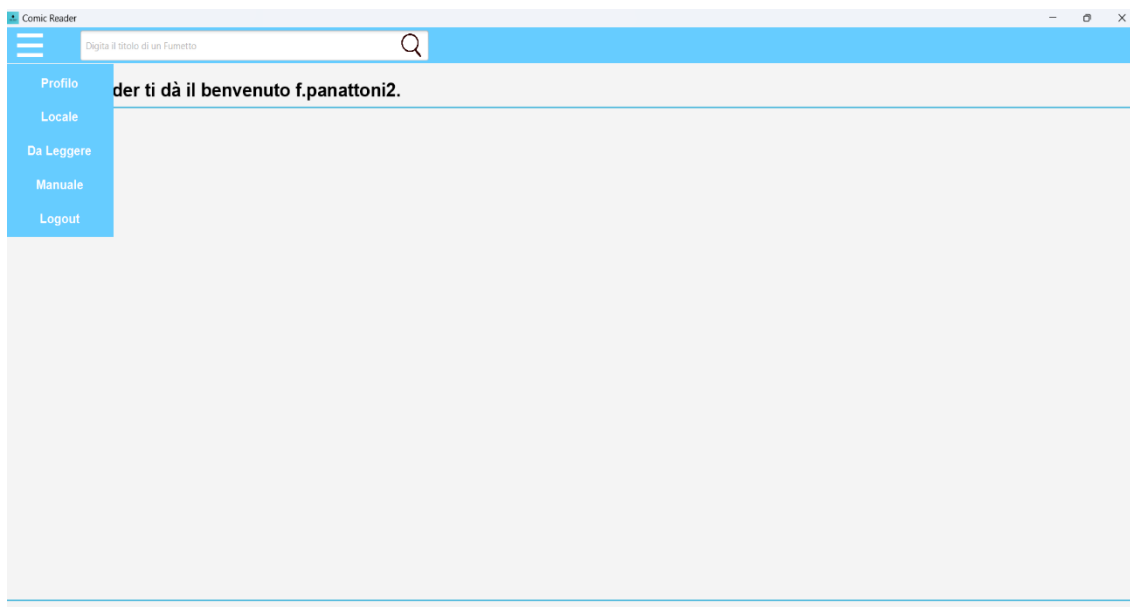
³ java.lang.String

3.2 Pagina Principale

Una volta fatto il Login o la Registrazione, ci troveremo nella Pagina Principale.



Qua possiamo scrivere sulla barra di ricerca, il nome del fumetto che cerchiamo, o possiamo utilizzare il menù a tendina. Al momento ci concentriamo sul menù a tendina, della barra di ricerca dei fumetti parleremo dopo.



Analizziamo tutte le voci del menù a tendina.

- Profilo: Permette di cambiare username, password e credenziali per il recupero password (domanda di sicurezza e risposta di sicurezza). Inoltre è possibile cambiare anche la modalità di lettura, spuntando l'opzione della lettura orientale per i manga. Attenzione: la lettura orientale (da destra verso sinistra) funziona solo per i fumetti che sono contrassegnati nel database come manga;
- Locale: Approfondita nel capitolo 2, sezione 2.5.2;
- Da Leggere: Ha una serie di tabelle che rappresentano la tabella dei fumetti finiti (il cui ultimo capitolo letto coincide con l'ultimo capitolo presente nel database), la tabella dei fumetti da finire e la tabella dei fumetti da leggere. In particolare in quest'ultima potremo inserire dei fumetti presenti nell'applicazione, che per qualsiasi motivo non possiamo leggere ora, ma che vorremo leggere in futuro. È possibile rimuovere una riga da qualsiasi tabella attraverso il click destro del mouse e cliccando sull'opzione "Rimuovi". Se leggi un fumetto, le tabelle si aggiornano automaticamente, quindi se leggi un fumetto messo nella tabella dei fumetti da leggere, verrà automaticamente spostato nella tabella dei fumetti in corso di lettura o nella tabella dei fumetti finiti se si è aperto l'ultimo capitolo.
- Manuale: C'è un manuale che spiega all'utente come funziona l'applicazione.
- Logout: Ti permette di tornare alla schermata di Login.

3.2.1 Dettagli Implementativi

I package che si occupano di questa parte sono "it.unipi.panattoni.client" che gestisce sia la Pagina Principale che l'intera applicazione e "it.unipi.panattoni.client.account" che gestisce le opzioni del menù a tendina.

Partendo da "it.unipi.panattoni.client":

- App.java: L'applicazione con il main. Si configura con JavaFX e apre home.fxml. Mette come titolo dell'applicazione "Comic Reader" e mostra come icona dell'applicazione il logo dell'applicazione;
- SessioneUtente.java: Permette di creare una sessione personalizzata all'interno dell'applicazione. Necessaria per l'aggiornamento dei dati del database. Viene implementata usando il Singleton Pattern;
- HomeController.java: Il controllore di home.fxml, ovvero della Pagina Principale dove è possibile aprire e chiudere il menù a tendina e consultare la barra di ricerca dove si può selezionare il fumetto da leggere.

Nel package "it.unipi.panattoni.client.account" abbiamo:

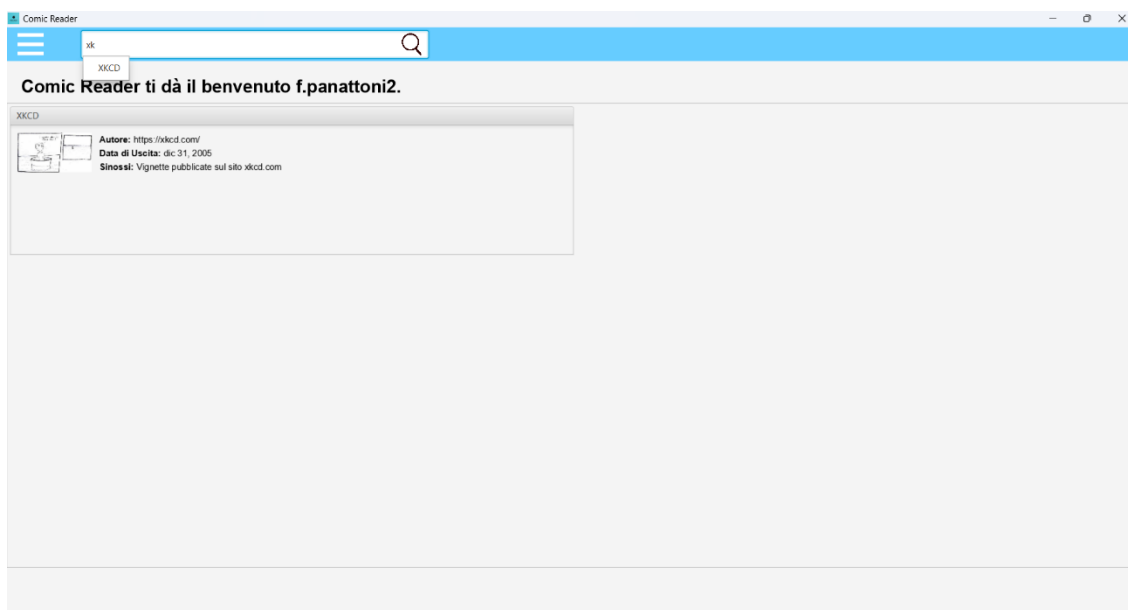
- ProfiloController.java: Il controllore di profilo.fxml, che gestisce la schermata dell'account dove è possibile cambiare username e tipo di lettura per manga. Si può accedere alla schermata di cambio password e di cambio credenziali;
- ChangePasswordController.java: Il controllore di changepwd.fxml (quello dell'account), che gestisce il cambio di password;
- ChangeSecurityController.java: Il controllore di changesecurity.fxml, che gestisce il cambio delle credenziali per il cambio password (Domanda di Sicurezza e Risposta di Sicurezza);
- LocaleController.java: Il controllore di locale.fxml, che gestisce l'aggiornamento delle informazioni riguardo ad un fumetto aggiunto localmente (si guardi il capitolo 2, sezione 2.5.2);
- LetturaController.java: Il controllore di lettura.fxml, che gestisce i dati riguardanti i fumetti finiti, in corso di lettura e da leggere in futuro;
- ManualeController.java: Il controllore di manuale.fxml, che gestisce il manuale dell'applicazione.

Il Logout è gestito da *HomeController*.

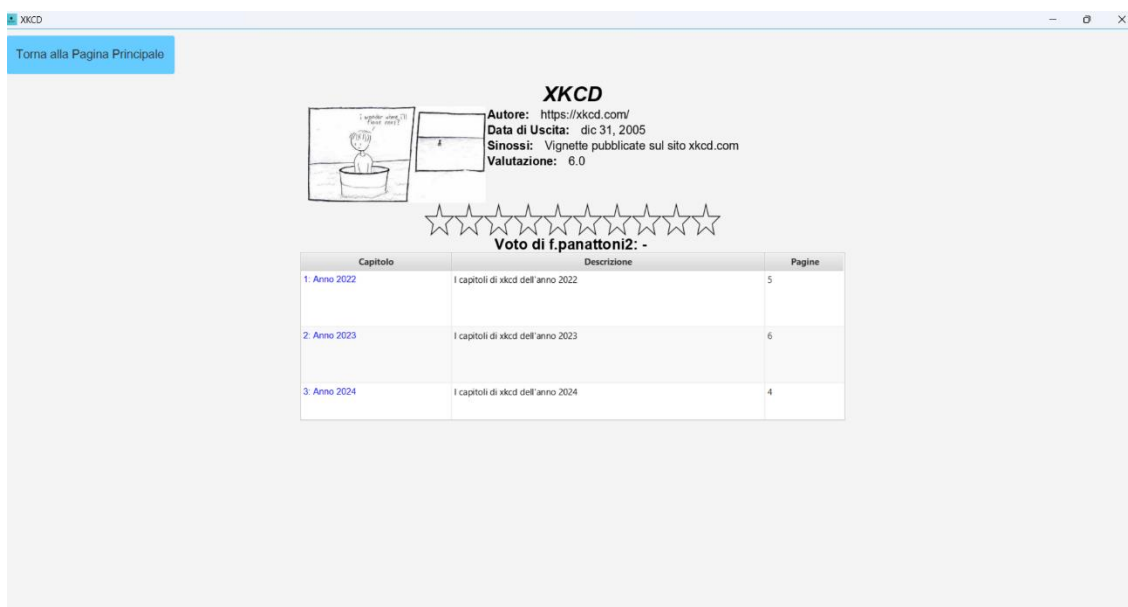
La differenza fra *ChangePasswordController* di account e *ChangePasswordController* di login è che il primo usa lo username di *SessioneUtente*, mentre il secondo utilizza una stringa che gli viene passata come username.

3.3 Lettura dei Fumetti

Dalla schermata principale, è possibile inserire il nome del fumetto desiderato nella barra di ricerca. Verrà visualizzato un blocco informativo con la copertina del fumetto. Cliccando su questo blocco, si accederà al menù di lettura del fumetto.



Dopo aver cliccato sul blocco informativo, verremo indirizzati nella schermata del fumetto, dove possiamo leggere uno dei capitoli. Da notare che il nome dell'applicazione è cambiato ed è diventato quello del fumetto scelto. Potremo provare ad inserire una votazione, ma ci apparirà un messaggio che ci dice che non è possibile perché ancora non abbiamo letto un capitolo. Clicchiamo sul titolo del primo capitolo.



Possiamo muoverci tra le pagine del fumetto attraverso le frecce. Una volta arrivati in fondo, sarà possibile infine passare al successivo capitolo senza tornare alla schermata del fumetto.



Inoltre è possibile zoomare la pagina con il click sinistro del mouse e de-zoomare con il click destro del mouse.

3.3.1 Dettagli Implementativi

Il package che si occupa di questa parte è “it.unipi.panattoni.client.fumetti”.

Di questo package fanno parte:

- Capitolo.java: Rappresenta i dati del capitolo di un fumetto nel sistema;
- Fumetto.java: Rappresenta i dati del fumetto nel sistema;
- Lettura.java: Rappresenta la relazione tra il fumetto e l'utente, ossia i fumetti letti effettivamente dall'utente;
- ComicController.java: Il controllore di comic.fxml, che gestisce la visualizzazione delle informazioni sui fumetti e sui capitoli, la valutazione sul fumetto e l'accesso al capitolo da leggere;
- ReadController.java: Il controllore di read.fxml, che gestisce la visualizzazione delle pagine del fumetto, l'aggiornamento della lettura nel database e lo scorrimento delle pagine.

La valutazione può essere espressa tramite un sistema di dieci stelle, permettendo di assegnare un punteggio da 1 a 10. Le stelle sono rappresentate da "Regioni" (Region) personalizzate attraverso il file "stelle.css".

4 Conclusione

In conclusione, Comic Reader dimostra come un'interfaccia facile da usare possa migliorare l'esperienza degli appassionati di fumetti e manga. Le sue funzioni, come il tracciamento delle letture e la personalizzazione delle collezioni digitali, sono progettate per rendere tutto più semplice. L'uso di tecnologie web moderne e un design intuitivo riflettono l'attenzione verso le esigenze degli utenti durante lo sviluppo del progetto. Sono sicuro che Comic Reader possa rispondere alle aspettative di chi cerca una piattaforma pratica e accessibile per gestire e esplorare il mondo dei fumetti.