

**Members of the Team:** Zacarias Vasco Chironda

Muino Gilberto Taquidir

TODO: INDEX HERE

# Abstract

The "Txopela" project aims to introduce a dedicated mobile application for booking motor taxis in Mozambique. Unlike existing services such as Yango, which cater to both motor taxis and traditional taxis, Txopela exclusively focuses on the motor taxi category. Through the application, users will have access to a convenient and efficient platform for booking motor taxi rides, enhancing accessibility and reliability in urban transportation. This project entails comprehensive software engineering processes, including requirements analysis, system design, development, testing, and deployment, to ensure the creation of a robust and user-friendly application tailored to the unique needs of the Mozambican market. By providing a specialized solution for motor taxi booking, Txopela aims to streamline the transportation experience for both riders and drivers, contributing to improved mobility and convenience in Mozambique.

### Requirement Analysis

###### System requirements

###### **Functional Requirements:**

###### User Registration: The app should allow users to create accounts with unique credentials.

###### Booking Management: Users should be able to book motor taxi rides seamlessly through the app.

###### GPS Integration: The app should utilize GPS functionality to identify user locations and provide accurate pickup and drop-off locations.

###### Payment Integration: Integration with payment gateways to facilitate secure and convenient payment transactions within the app.

###### Ride Tracking: Users should be able to track the location of their booked motor taxi in real-time.

###### Rating and Review System: Allow users to rate and provide feedback on their ride experience.

###### Notifications: The app should send timely notifications regarding booking confirmation, driver arrival, and other relevant updates.

###### Multilingual Support: Support for multiple languages to cater to a diverse user base in Mozambique.

###### Emergency Assistance: Include features for users to request emergency assistance or report safety concerns during their ride.

###### **End User Requirements:**

###### User Registration: End users should be able to create accounts and provide necessary information.

###### Booking Management: End users should have the ability to book motor taxi rides, view ride history, and manage bookings.

###### GPS Integration: The app should use GPS functionality to identify user locations and provide accurate pickup and drop-off locations.

###### Payment Integration: End users should be able to make secure payments for their rides through the app.

###### Rating and Review System: End users should have the ability to rate drivers and provide feedback on their ride experience.

###### Notifications: End users should receive notifications about booking confirmation, driver arrival, and other relevant updates.

###### Multilingual Support: Support for multiple languages to cater to the diverse user base in Mozambique.

###### **Driver System Requirements:**

###### Driver Registration: Drivers should be able to register with the platform, provide necessary documentation, and undergo verification.

###### Ride Management: Drivers should have the ability to accept or decline ride requests, view ride details, and manage their availability.

###### GPS Integration: The app should provide navigation assistance to drivers for efficient route planning and navigation.

###### Payment Processing: Drivers should receive payments for completed rides and have access to payment history.

###### Rating System: Drivers should receive ratings and feedback from users based on their performance.

###### Notifications: Drivers should receive notifications about new ride requests, ride cancellations, and other relevant updates.

###### **Admin System Requirements:**

###### User Management: Admins should have the ability to manage user accounts, including registration, verification, and account suspension if necessary.

###### Driver Management: Admins should be able to onboard new drivers, review driver documentation, and manage driver accounts.

###### Ride Monitoring: Admins should have access to real-time ride data, including active rides, completed rides, and ride histories.

###### Payment Management: Admins should oversee payment processing, including transaction monitoring, dispute resolution, and payout management.

###### Reporting and Analytics: Admins should have access to reporting tools and analytics to track app usage, user behavior, and driver performance.

###### Platform Configuration: Admins should be able to configure app settings, including pricing, service areas, and promotional offers.

###### **Non-Functional Requirements:**

###### Performance: Ensure that the app is responsive and performs efficiently, even under heavy user load.

###### Security: Implement robust security measures to protect user data and payment transactions.

###### Reliability: The app should be dependable, with minimal downtime and reliable service availability.

###### Compatibility: Compatibility with a wide range of mobile devices and operating systems (iOS and Android).

###### Usability: User interface design should be intuitive and user-friendly, catering to users with varying levels of technical proficiency.

###### Scalability: The system should be designed to accommodate future growth and increasing user demand.

###### Accessibility: Ensure accessibility features for users with disabilities to make the app inclusive.

###### Compliance: Compliance with local regulations and legal requirements for mobile applications operating in Mozambique.

###### **END USER REQUIREMENTS**

###### Security: End users' personal and payment information must be securely stored and transmitted.

###### Performance: The app should respond promptly to user actions, ensuring a smooth and efficient booking experience.

###### Reliability: The app should be available and reliable, with minimal downtime and service interruptions.

###### Compatibility: The app should be compatible with a wide range of mobile devices and operating systems.

###### Usability: The app interface should be intuitive and user-friendly, catering to users with varying levels of technical proficiency.

###### Accessibility: The app should be accessible to users with disabilities, complying with accessibility standards.

###### Compliance: The app should comply with relevant regulations and legal requirements for mobile applications.

###### **Driver System Requirements:**

###### Security: Driver accounts and payment transactions must be secure and protected from unauthorized access.

###### Performance: The app should perform efficiently for drivers, ensuring quick response times and minimal lag.

###### Reliability: The app should be reliable for drivers, with dependable service availability and minimal downtime.

###### Compatibility: The app should be compatible with the devices and operating systems commonly used by drivers.

###### Usability: The app interface for drivers should be intuitive and easy to use, facilitating efficient ride management.

###### Scalability: The app should be scalable to accommodate a growing number of drivers and increasing demand for rides.

###### Compliance: The app should comply with relevant regulations and legal requirements applicable to driver services.

###### **Admin System Requirements:**

###### Security: Admin accounts and access to sensitive data must be secured to prevent unauthorized access.

###### Performance: The admin interface should perform efficiently, allowing admins to manage the platform without delays or glitches.

###### Reliability: The admin system should be reliable, with minimal downtime and service interruptions.

###### Usability: The admin interface should be user-friendly, with intuitive navigation and clear presentation of data.

###### Compliance: The admin system should comply with relevant regulations and legal requirements for data management and administration.

|  |  |  |
| --- | --- | --- |
| ID  Requisito | REQUISITO | TIPO |
| R1F | L’utilizzo del servizio è possibile solo previa registrazione degli  utenti | Funzionale |
| R2F | In fase di registrazione è necessaria la distinzione tra lettore e scrittore | Funzionale |
| R3F | Per la registrazione del lettore è necessario specificare nome, cognome, username e password | Funzionale |
| R4F | Lo scrittore deve provare la sua identità durante la registrazione fornendo un documento di identità valido | Funzionale |
| R5F | Gli utenti devono essere identificati tramite username | Funzionale |
| R6F | Gli utenti devono essere autenticati tramite username e password | Funzionale |
| R7F | Il lettore e lo scrittore possono effettuare pagamenti tramite carta di credito | Funzionale |
| R8F | Lo scrittore può caricare il titolo, l’indice e il testo integrale del  proprio libro | Funzionale |
| R9F | Lo scrittore può fornire le opere integrali, in tal caso il libro sarà disponibile ai lettori in formula “trial” | Funzionale |
| R10F | Lo scrittore può sponsorizzare i propri libri previo pagamento | Funzionale |
| R11F | Il sistema deve rimuovere automaticamente il libro tra gli sponsorizzati dopo 7 giorni | Funzionale |
| R12F | Il lettore usufruisce della piattaforma o nella versione standard o nella versione Premium | Funzionale |
| R13F | Il lettore può cercare libri | Funzionale |
| R14F | Il lettore può aggiungere e rimuovere più libri dalla collezione  “letture in corso” | Funzionale |
| R15F | Il lettore può selezionare come letti i capitoli di un particolare libro | Funzionale |

|  |  |  |
| --- | --- | --- |
| R16F | Il lettore può valutare il capitolo letto con una valutazione da 1 a 5 stelle | Funzionale |
| R17F | Il lettore può assegnare al capitolo letto uno stato d’animo  rappresentato da un’emoji | Funzionale |
| R18F | Il lettore può rilasciare un commento relativo ad un capitolo letto | Funzionale |
| R19F | Il lettore può segnalare un commento | Funzionale |
| R20F | Il lettore può leggere libri “trial” inseriti negli slot direttamente  dalla piattaforma | Funzionale |
| R21F | Il lettore può aggiungere libri in forma “trial” allo slot | Funzionale |
| R22F | Il lettore può usufruire della versione Premium previo pagamento | Funzionale |
| R23F | Le funzionalità disponibili per il lettore sono presenti anche per lo scrittore, ma non viceversa | Funzionale |
| R24F | L’amministratore possiede tutti i privilegi, tra cui la  creazione/eliminazione di account assistenza e le funzionalità  dell’assistenza | Funzionale |
| R25F | L’assistenza può bloccare e sbloccare utenti della piattaforma | Funzionale |
| R26F | L’assistenza può rimuovere commenti | Funzionale |
| R27F | L’assistenza può confermare o rifiutare la richiesta di registrazione di uno scrittore | Funzionale |
| R28F | Gli slot degli utenti vengono liberati automaticamente alla scadenza del tempo disponibile per la lettura del Trial | Funzionale |
| R1NF | Velocità di ricerca del libro | Non funzionale |
| R2NF | Semplicità di navigazione nell’applicazione | Non funzionale |
| R3NF | Velocità di registrazione degli account | Non funzionale |

|  |  |  |
| --- | --- | --- |
| R4NF | Velocità di inserimento del libro con tutti i dati caratterizzanti | Non funzionale |
| R5NF | Protezione dei dati sensibili degli account per mezzo di meccanismi di cifratura | Non funzionale |
| R6NF | Periodico backup dei dati | Non funzionale |
| R7NF | Un commento non può essere modificato | Non funzionale |
| R8NF | Un libro non può essere rivendicato da nessuno scrittore se  non dall’autore legittimo | Non funzionale |
| R9NF | Non possono essere inseriti più libri contemporaneamente  all’interno di uno slot | Non funzionale |
| R10NF | Ogni slot deve poter contenere solo libri disponibili in forma  “trial” | Non funzionale |
| R11NF | Nella versione standard, dal momento dell’inserimento del  libro nello slot, questo sarà disponibile integralmente per un  periodo massimo di 3 giorni | Non funzionale |
| R12NF | Nella versione Premium, al momento dell’inserimento del libro  nello slot, questo sarà disponibile integralmente per un periodo massimo di 14 giorni | Non funzionale |
| R13NF | Nella versione standard il lettore usufruisce di 1 slot | Non funzionale |
| R14NF | Nella versione Premium il lettore usufruisce di 3 slot | Non funzionale |
| R15NF | Qualsiasi transazione economica (sponsorizzazione o passaggio a Premium) avviene tramite PayPal | Non Funzionale |
| R16NF | L’applicazione dovrà essere rilasciata sottoforma di file .war | Non Funzionale |

## Analisi del dominio

##### Glossario

|  |  |  |
| --- | --- | --- |
| VOCE | DEFINIZIONE | SINONIMI |
| Account | Qualunque entità che interagisca con il software, sia essa un lettore, uno scrittore, l’amministratore o  l’assistenza. | Attore |
| Amministratore | Account che possiede tutti i privilegi. |  |
| Assistenza | Account che si occupa di eliminare tutti i commenti con contenuto improprio e non educato e  bloccare/sboccare utenti. |  |
| Capitolo | Ciascuna delle parti più o meno ampie in cui si divide un libro. |  |
| Collezione "letture in corso" | Insieme di libri di cui un utente ha letto almeno un capitolo, non ancora terminati o non ancora rimossi  volontariamente dall'insieme stesso. |  |
| Commento | Giudizio espresso in relazione ad un capitolo. | Recensione |
| Credenziali | Username e password. |  |
| Documento di identità | Documento con la finalità di dimostrare l'identità personale del suo titolare. |  |
| Emoji | Piccola icona a colori usata nella comunicazione elettronica per esprimere un concetto o  un'emozione. | Emoticon, Reazione |
| Indice | Elenco dei nomi dei capitoli che distinguono le varie  parti in cui l’opera è suddivisa. |  |
| Lettore | Utente che usufruisce dell'applicazione principalmente per la lettura e il commento di libri. |  |
| Libro | Entità logica che rappresenta l'opera letteraria come unione di struttura e autore | Opera |
| Login | Operazione necessaria per identificare e autenticare un utente. | Autenticazione, Accesso |
| Opera integrale | Intero contenuto del libro in formato digitale. | Testo integrale |

|  |  |  |
| --- | --- | --- |
| Pagamento | Trasferimento di una somma di denaro. |  |
| Password | Stringa di riconoscimento fornita dall'utente al software per poter accedere. |  |
| PayPal | Servizio che consente di pagare, inviare denaro e  accettare pagamenti in modo rapido, semplice e sicuro |  |
| Registrazione | Fase in cui il nuovo utente deposita informazioni personali. |  |
| Scrittore | Utente che gode delle funzionalità disponibili per i  lettori che al contempo ha la possibilità di caricare proprie opere. | Autore |
| Segnalazione | Avviso inoltrato dall’utente all’assistenza per  notificare un commento inappropriato. |  |
| Slot | Spazio logico che può contenere un testo integrale. |  |
| Sponsorizzazione | Finanziamento da parte dello scrittore al fine di promuovere e far pubblicità al titolo di un proprio  libro. | Pubblicità, Promozione |
| Stella | Unità di gradimento. |  |
| Struttura | Insieme costituito da titolo e indice di un libro. |  |
| Titolo | Nome o frase che serve a individuare un libro. |  |
| Trial | Modalità di fruizione del libro in forma integrale che ne permette la lettura con delle limitazioni  temporali. | Prova |
| Username | Stringa alfanumerica. |  |
| Utente | Un attore: scrittore o lettore. |  |
| Utente attivo | Un utente che può essere autenticato e usufruire dei  servizi dell’applicazione. | Utente attivo |
| Utente bloccato | Un utente che non deve essere autenticato.  L’assistenza deve poter bloccare gli utenti. |  |

|  |  |  |
| --- | --- | --- |
| Valutazione | Determinazione del gradimento di un capitolo specificando un voto da 1 a 5 stelle. |  |
| Versione Premium | Versione dell'applicazione che presenta funzionalità aggiuntive e standard migliorate. | Versione a pagamento |
| Versione standard | Versione dell'applicazione con funzionalità base e  identiche per tutti gli utenti che non hanno effettuato l'abbonamento alla versione Premium. | Versione gratuita, di base |

## Analisi dei requisiti

##### Casi d’uso

Ricerca



GestioneCollezione

GestioneTrial

GestioneCapitolo

Registrazione

<<extend>>

<<extend>>

<<extend>>

<<extend>>

PassaggioAPremium

<<extend>>

GestioneLettore RimozioneTrial

Lettore

Login

<<includ e>>

<<event>>

ScadenzaTrial

<<includ e>>

<<includ e>>

Gestione Amministratore

GestioneScrittore

<<includ e>>

Amministratore

Scrittore

<<extend>>

Inizio

Sponsorizzazione

Gestione

Assistenza

Assistenza

FineSponsorizzazione

<<event>>

Scadenza Sponsorizzazione

##### Scenari

|  |  |
| --- | --- |
| Titolo | Login |
| Descrizione | Permette di accedere al sistema |
| Attori | Amministratore, Assistenza, Scrittore, Lettore |
| Relazioni | GestioneAmministratore, GestioneAssistenza, GestioneScrittore,  GestioneLettore |
| Precondizioni | L’attore deve essere registrato nel sistema |
| Postcondizioni | L’attore è autenticato nel sistema |
| Scenario principale | 1. Il sistema presenta all’attore una maschera per l’inserimento delle   credenziali di accesso: username e password   1. L’attore inserisce i dati del proprio account e conferma 2. Il sistema verifica le credenziali 3. Dopo l’autenticazione viene mostrata la schermata consona al tipo   di attore |
| Scenari alternativi | 1. Credenziali inserite non riconosciute:    1. Il sistema notifica l’attore con un avviso consono    2. Il sistema ripresenta all’attore la maschera di inserimento credenziali 2. Account bloccato:    1. Il sistema notifica l’attore con un avviso relativo allo stato   dell’account   * 1. Il sistema ripresenta all’attore la maschera di inserimento   credenziali   1. La registrazione dello Scrittore non è stata ancora accettata    1. Il sistema notifica che l’account deve ancora essere attivato    2. Il sistema ripresenta all’attore la maschera di inserimento   credenziali |
| Requisiti non  funzionali | R5NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneAmministratore |
| Descrizione | Permette di usufruire delle funzionalità dell’Assistenza ed ha il controllo  sugli account dell’Assistenza stessa |
| Attori | Amministratore |
| Relazioni | Login |
| Precondizioni | Essere in possesso delle credenziali per accedere come amministratore |
| Postcondizioni |  |
| Scenario principale | 1. Login |

|  |  |
| --- | --- |
|  | 1. Il sistema presenta una schermata che permette   all’Amministratore di eseguire le funzionalità disponibili   1. L’Amministratore prosegue con la creazione o con l’eliminazione   di account Assistenza o con una delle funzioni disponibili per  l’Assistenza stessa. |
| Scenari alternativi | 1. L’amministratore cerca di creare un account Assistenza con un   username esistente.   * 1. Viene notificato un messaggio di errore   2. L’amministratore viene riportato sulla propria home   facendolo ripartire dal punto 2 |
| Requisiti non  funzionali | R2NF, R3NF, R5NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneAssistenza |
| Descrizione | Gestione di tutte le funzionalità disponibili all’Assistenza e, di  conseguenza, all’Amministratore |
| Attori | Assistenza |
| Relazioni | Login |
| Precondizioni | L’Amministratore deve aver creato delle credenziali per account  Assistenza |
| Postcondizioni |  |
| Scenario principale | 1. Login 2. Il sistema presenta una schermata che permette all’Assistenza di   eseguire le sue funzionalità   1. L’Assistenza prosegue con l’eliminazione di un commento, il (s)blocco di un account o l’accettazione di una richiesta di   registrazione di uno scrittore dopo l’analisi della stessa |
| Scenari alternativi | A) L’identità dello scrittore che ha richiesto la registrazione non può  essere validata:  3. L’Assistenza rifiuta la richiesta di registrazione |
| Requisiti non  funzionali | R2NF, R5NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneScrittore |
| Descrizione | Gestione di tutte le funzionalità di caricamento e sponsorizzazione dei  libri |
| Attori | Scrittore |
| Relazioni | Login |
| Precondizioni | Lo Scrittore deve essersi registrato sulla piattaforma |
| Postcondizioni |  |
| Scenario principale | 1. Login 2. Il sistema presenta all’utente la schermata principale particolare per l’account di tipo scrittore 3. Lo Scrittore avrà la possibilità di eseguire tutte le funzionalità disponibili per il Lettore e inoltre potrà caricare titolo, indice e testo integrale di un proprio libro, sponsorizzare una propria opera   o rimuovere il testo integrale. |
| Scenari alternativi | 1. Il titolo che lo Scrittore vuole caricare è già presente tra i suoi libri:    1. Il sistema impedisce il caricamento di quel libro nel sistema    2. Il sistema notifica lo Scrittore con un avviso consono    3. Lo Scrittore ha di nuovo la possibilità di inserire un proprio libro 2. Se il testo integrale non è stato precedentemente caricato    1. Lo scrittore cerca di rimuoverlo    2. Un messaggio di errore viene notificato    3. Lo Scrittore ha di nuovo la possibilità di rimuovere un proprio testo |
| Requisiti non  funzionali | R2NF, R4NF, R5NF, R8NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | InizioSponsorizzazione |
| Descrizione | Permette allo Scrittore di sponsorizzare un proprio libro mettendolo in  evidenza nella sezione di ricerca |
| Attori | Scrittore |
| Relazioni | GestioneScrittore |
| Precondizioni | L’Autore del libro deve aver caricato il proprio libro da sponsorizzare |
| Postcondizioni | Il libro comparirà in evidenza a seguito di ricerche dei lettori |
| Scenario principale | 1. Il sistema presenta allo Scrittore la propria home che include tutti i libri caricati dallo stesso 2. Lo Scrittore sceglie un libro da sponsorizzare 3. Il sistema presenta una schermata di pagamento |

|  |  |
| --- | --- |
|  | 1. Lo Scrittore inserisce i dati della propria carta di pagamento e conferma 2. Il sistema verifica la correttezza dei dati e procede con la transazione 3. Il sistema imposta il libro come sponsorizzato |
| Scenari alternativi | 1. I dati della carta di pagamento sono errati o invalidi    1. Il sistema controlla i dati e non riesce a validarli    2. Il sistema presenta allo scrittore la maschera del punto 1. |
| Requisiti non  funzionali | R2NF, R5NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | FineSponsorizzazione |
| Descrizione | Definisce il termine della sponsorizzazione di un certo libro |
| Attori | ScadenzaSponsorizzazione «event» |
| Relazioni |  |
| Precondizioni | Il libro deve risultare sponsorizzato |
| Postcondizioni | Il libro associato alla sponsorizzazione non viene più mostrato in evidenza |
| Scenario principale | 1. Si verifica l’evento FineSponsorizzazione 2. Il sistema rimuove il titolo tra quelli sponsorizzati nella sezione di ricerca |
| Scenari alternativi |  |
| Requisiti non  funzionali |  |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneLettore |
| Descrizione | Gestione di tutte le funzionalità di commento, recensione e lettura libri |
| Attori | Lettore |
| Relazioni | Login |
| Precondizioni | Il Lettore deve essere in possesso delle proprie credenziali d’accesso |
| Postcondizioni |  |
| Scenario principale | 1. Login 2. Il Lettore può cercare libri, accedere alle funzionalità di GestioneCollezione, di GestioneTrial e, dopo aver letto un capitolo, anche di GestioneCapitolo.   In più può passare a Premium, nel caso sia un utente Standard. |
| Scenari alternativi |  |
| Requisiti non  funzionali | R2NF |

|  |  |
| --- | --- |
| Punti aperti |  |
| Titolo | Registrazione |
| Descrizione | Permette di registrarsi alla piattaforma |
| Attori | Lettore |
| Relazioni |  |
| Precondizioni | L’utente non deve risultare attualmente registrato nella piattaforma |
| Postcondizioni | L’utente è in possesso delle credenziali per l’accesso alla piattaforma |
| Scenario principale | 1. L’Utente visita la pagina di registrazione nuovo Utente 2. La piattaforma presenta all’Utente la possibilità di registrarsi   come Lettore o Scrittore   1. Nel caso di Lettore: l’Utente inserisce nome, cognome, username e password   Nel caso di Scrittore: l’Utente inserisce, oltre ai dati sopra citati,  un documento d’identità   1. L’Utente deve confermare in modo esplicito la conclusione della registrazione 2. Il sistema controlla l’univocità dello Username e se verificata ne memorizza tutti i dati associati 3. Viene visualizzato un messaggio di conferma di avvenuta registrazione e nel caso dello scrittore, viene aggiunto un messaggio di invito ad attendere la verifica e l’abilitazione definitiva dell’account da parte dell’assistenza. 4. Il sistema mostra la schermata di login |
| Scenari alternativi | A) Username non univoco  6. Se lo Username è stato già associato ad un altro utente deve essere visualizzata nuovamente la sezione di registrazione in aggiunta ad un messaggio di errore di  mancata registrazione, ripartendo così dal punto 1 |
| Requisiti non  funzionali | R3NF, R5NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | Ricerca |
| Descrizione | Permette all’utente di cercare il titolo di un’opera |
| Attori | Lettore |
| Relazioni | GestioneLettore |
| Precondizioni | Il Lettore deve essere in possesso delle proprie credenziali d’accesso |
| Postcondizioni | Viene mostrato all’Utente il risultato della ricerca |

|  |  |
| --- | --- |
| Scenario principale | 1. L’utente effettua la ricerca di un titolo 2. Ottiene una lista con le opere trovate con quel titolo e alcuni titoli sponsorizzati |
| Scenari alternativi | A) Il titolo cercato non è presente  2. Viene mostrato un messaggio che comunica l’assenza di quel  libro |
| Requisiti non  funzionali | R1NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneCollezione |
| Descrizione | Permette all’Utente di selezionare e visualizzare il libro che può essere  aggiunto o rimosso dalla collezione “letture in corso” |
| Attori | Lettore |
| Relazioni | GestioneLettore |
| Precondizioni | Il Lettore deve essere in possesso delle proprie credenziali d’accesso |
| Postcondizioni |  |
| Scenario principale | 1. Il Lettore ha raggiunto la schermata principale fornita dal sistema 2. Il Lettore interagisce con il sistema per:    * aggiungere un libro alla collezione “letture in corso”    * visualizzare la collezione “letture in corso”    * rimuovere un libro dalla collezione “letture in corso” 3. Il sistema esegue l’operazione selezionata dall’Utente 4. Il sistema restituirà al Lettore:    * in caso di selezione, la struttura del libro    * in caso di aggiunta o rimozione, la collezione aggiornata |
| Scenari alternativi | A)  2. L’Utente cerca di aggiungere alla collezione un titolo già presente La richiesta viene ignorata  B)  2. L’Utente cerca di rimuovere dalla collezione un titolo già presente. La richiesta viene ignorata |
| Requisiti non  funzionali | R2NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneCapitolo |
| Descrizione | Permette di interagire con le sezioni relative al singolo capitolo e di  usufruire della sezione commenti associata |
| Attori | Lettore |
| Relazioni | GestioneLettore |
| Precondizioni | Il capitolo è stato selezionato come letto dall’Utente |
| Postcondizioni | Vengono memorizzate le valutazioni dell’utente ed eventuali commenti. |
| Scenario principale | 1. L’Utente visualizza una schermata in cui sono disponibili le funzionalità per esprimere un voto da 1 a 5 stelle, selezionare un’emoticon tra quelle proposte, scrivere segnalare commenti 2. L’Utente interagisce con le varie sezioni 3. L’Utente avrà terminato una o più delle seguenti operazioni: selezionato un numero di stelle compreso tra 1 e 5, selezionato   un’emoticon, scritto o segnalato un commento. |
| Scenari  alternativi |  |
| Requisiti non  funzionali | R7NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | RimozioneTrial |
| Descrizione | Il Sistema rimuove automaticamente il libro dallo Slot del Lettore |
| Attori | ScadenzaTrial «event» |
| Relazioni |  |
| Precondizioni | Il Libro deve essere stato inserito nello Slot |
| Postcondizioni | Il Libro non è più presente nell’apposito Slot |
| Scenario  principale | 1. Si è verificato l’evento di ScadenzaTrial 2. Il Sistema rimuove il Libro dallo Slot |
| Scenari alternativi |  |
| Requisiti non  funzionali |  |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneTrial |
| Descrizione | L’Utente può inserire un Libro in formato “trial” in uno dei suoi Slot e  visualizzarne il testo integrale |
| Attori | Lettore |
| Relazioni | GestioneLettore |
| Precondizioni | Il Libro deve essere stato inserito in formato “trial” dallo Scrittore e, in caso  di aggiunta, deve essere presente almeno uno Slot libero |
| Postcondizioni | In caso di aggiunta di un “trial”, uno degli slot liberi viene occupato |
| Scenario principale | 1. Il Lettore ha la possibilità di selezionare il Libro in formato “trial” da   aggiungere oppure di visualizzare la scheda di un Libro già inserito   1. A seconda dell’azione il Libro viene aggiunto o visualizzato |
| Scenari alternativi | 1. Tutti gli Slot sono occupati:    1. Il Libro non viene aggiunto    2. Un messaggio notifica l’errore al Lettore 2. Se lo Scrittore ha eliminato il testo integrale del Libro    1. Il libro non può essere aperto o aggiunto |
| Requisiti non  funzionali | R9NF, R10NF, R11NF, R12NF, R13NF, R14NF |
| Punti aperti |  |

## Analisi del rischio

##### Tabella Valutazione dei Beni

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bene | Valore | Esposizione | | |
| Sistema Informativo | Alto.  Supporto alla gestione della piattaforma | Alto.  Perdita di immagine e finanziaria. Costo di tempo per il rispristino dei dati da backup. | | |
| Informazioni sugli utenti | Medio.  Nome, cognome e  credenziali degli utenti | Medio.  Perdita di immagine e costo per il rispristino di  backup | | |
| Informazioni dell’Assistenza | Alto. Credenziali dell’Assistenza | Alto.  Perdita di immagine dovuta al fatto che gli utenti potrebbero essere bloccati dal possessore delle credenziali di assistenza | | |
| Informazioni  dell’Amministratore | Alto.  Credenziali  dell’Amministratore | Alto.  Perdita della gestione dell’applicazione. | | |
| DataBase dei libri | Medio. | Medio.  Perdita di immagine e finanziaria poiché gli scrittori potrebbero contestare la diffusione di opere di loro proprietà che avevano reso disponibili per un periodo limitato in formato trial.  Costo di tempo per il rispristino dei dati da backup | | |
| Tabella Minacce e Controlli | | | | |
| Minaccia | Probabilità | | Controllo | Fattibilità |
| Furto di identità Amministratore | Bassa: username e password non sono facili  da identificare | | Log degli accessi, accesso da macchine  sicure | Basso costo |
| Furto di identità Assistenza | Bassa: username e password non sono facili da identificare | | Log degli accessi | Basso costo |
| Furto di identità Utente | Medio-Alta: ingegneria sociale | | Log degli accessi | Medio-alto costo |
| DoS | Alta | | Progettazione adeguata, controllo e limitazione degli accessi | Basso costo: impossibile prevenire un DoS |

|  |  |  |  |
| --- | --- | --- | --- |
| Intercettazione comunicazione e pagamento: Man in the Middle | Medio-Alta | Cifratura e log delle comunicazioni e del pagamento | Medio-alto: la cifratura è ibrida: chiavi asimmetriche e simmetriche |

##### Analisi Tecnologica della Sicurezza

|  |  |
| --- | --- |
| Tecnologia | Vulnerabilità |
| Cifratura delle comunicazioni | Utilizzo di una cifratura ibrida, simmetrica e asimmetrica, con le loro relative vulnerabilità:  Cifratura Simmetrica:   * Tempo di vita della chiave * Più informazioni vengono cifrate con la stessa chiave, più materiale è offerto per l’analisi del testo a un attaccante * Memorizzazione della chiave * Lunghezza della chiave Cifratura Asimmetrica: * Lunghezza della chiave * Memorizzazione della chiave privata. L’utilizzo di   cifrature standard non recenti potrebbero causare  vulnerabilità ulteriori. |
| Autenticazione tramite credenziali | * Social engineering: un utente potrebbe essere vittima di un attacco di social engineering, volto al furto delle credenziali. * Password cracking: attraverso metodologie di   cracking delle password, l’attaccante risale alla  password di un utente o amministratore di sistema.   * Negligenza: l’utente volontariamente o per sbaglio   fornisce le credenziali a terzi.   * Password banali |
| Architettura Client/Server | * Attacco DDoS/DoS: l’obiettivo dell’attacco è quello   di negare il servizio offerto agli utenti.   * Attacco Man in the Middle: un malintenzionato potrebbe riuscire a sottrarre informazioni o effettuare operazioni illecite. * Sniffing della comunicazione: un malintenzionato potrebbe riuscire ad intercettare una comunicazione e a sottrarne informazioni. * Deadlock: il servitore potrebbe trovarsi in una situazione di stallo generata da una richiesta ciclica. * Esposizione dei client: i client sono esposti, dato   che devono connettersi al servitore. |

##### Security Use Case e Misuse Case

Ricerca



GestioneCollezione

GestioneTrial

GestioneCapitolo

Registrazione

<<extend>>

<<extend>>

<<extend>>

<<extend>>

PassaggioAPremium

<<extend>>

GestioneLettore RimozioneTrial

Lettore

Login

<<includ e>>

<<event>>

ScadenzaTrial

<<includ e>>

<<includ e>>

Gestione

Amministratore

GestioneScrittore

<<includ e>>

Amministratore

Scrittore

<<extend>>

Inizio

Sponsorizzazione

Gestione

Assistenza

Assistenza

FineSponsorizzazione

<<event>>

Scadenza

Sponsorizzazione

Disponibilità

(Security)

Garantire Protezione

(Security)

Controllo Accesso

(Security)

DoS

(misuse)

Frode

(misuse)

ManInTheMiddle

(misuse)

Sniffing

(misuse)

FurtoCredenziali

(misuse)

Hacker Truffatore

|  |  |  |
| --- | --- | --- |
| *Titolo* | ControlloAccesso | |
| *Descrizione* | Gli accessi alla piattaforma devono essere controllati | |
| *Misuse case* | FurtoCredenziali, Sniffing | |
| *Precondizioni* | L’Attaccante ha i mezzi per scoprire almeno lo username di account di  qualsiasi tipo | |
| *Postcondizioni* | L’applicazione impossibilita temporaneamente l’accesso dell’utente e  notifica un tentativo di accesso fraudolento. | |
| *Scenario principale* | *Sistema* | *Attaccante* |
|  | Dopo aver individuato degli username tenta di accedere al sistema inserendo password  tramite un attacco con dizionario |
| Controlla che le credenziali immesse siano corrette, se non lo sono, dopo un certo numero di  tentativi, blocca l’accesso |  |
| *Scenario di un attacco avvenuto con successo* | *Sistema* | *Attaccante* |
|  | Attacco con dizionario riuscito |
| Il Sistema controlla le credenziali immesse e consente l’accesso al  sistema |  |
|  |  | Naviga tra le maschere del sistema e cerca di carpire più  informazioni possibili |
|  | Il Sistema scrive nel log tutte le  operazioni eseguite dall’utente |  |
|  | Il Sistema monitora periodicamente il log alla ricerca di pattern di accesso atipici e se rilevati notifica un accesso  fraudolento |  |

|  |  |  |
| --- | --- | --- |
| *Titolo* | GarantireProtezione | |
| *Descrizione* | Durante la fase di pagamento le transazioni devono essere monitorate  e validate | |
| *Misuse case* | Sniffing, ManInTheMiddle, Frode | |
| *Precondizioni* | 1. L’Attaccante ha i mezzi per intercettare le transazioni nel Sistema 2. L’Attaccante ha i mezzi per modificare le transazioni 3. L’Attaccante ha i mezzi per spedire la transazione modificata al destinatario | |
| *Postcondizioni* | Il Sistema notifica un tentativo di frode | |
| *Scenario principale* | *Sistema* | *Attaccante* |
| Monitora e protegge le transazioni  economiche e salva i dati non sensibili nei log |  |
|  | Intercetta una transazione che  sta avvenendo |
|  | Non riesce a decifrare i dati o a rimuovere i meccanismi di  protezione |
|  | Genera una transazione fraudolenta e tenta di inviarla al  destinatario |
| Si accorge che la transazione non è  pertinente e segnala un tentativo di frode |  |
| *Scenario di un attacco avvenuto con successo* | *Sistema* | *Attaccante* |
| Monitora e protegge le transazioni economiche e salva i dati non  sensibili nei log |  |
|  | Intercetta una transazione che  sta avvenendo |
|  | Riesce a rimuovere i meccanismi  di sicurezza |
|  | Modifica i dati, poi riapplica i  meccanismi di sicurezza e ritrasmette in dati |
| Elabora i dati e procede con la transazione salvando i dati non  sensibili nei log |  |
| Monitora  periodicamente i log alla ricerca di pattern atipici e se  rilevati notifica un accesso  fraudolento |  |

|  |  |  |
| --- | --- | --- |
| *Titolo* | Disponibilità | |
| *Descrizione* | Il Sistema deve essere sempre disponibile in tutte le sue funzionalità | |
| *Misuse case* | DoS | |
| *Relazioni* | GestioneLettore, GestioneScrittore | |
| *Precondizioni* | L’attaccante ha la potenza necessaria per mettere in azione questo tipo  di attacco in modo efficace | |
| *Postcondizioni* | Il Sistema notifica quando il flusso di richieste è fuori dall’ordinario e  attua delle misure di sicurezza | |
| *Scenario principale* | *Sistema* | *Attaccante* |
| Il Sistema ha i meccanismi di  protezione attivi |  |
|  | Il malintenzionato mette in  azione l’attacco |
| Il Sistema notifica il tentativo di DoS  mentre cerca di ridurre l’efficacia dell’attacco |  |
|  | L’attacco non ha successo |
| *Scenario di un attacco avvenuto con successo* | *Sistema* | *Attaccante* |
| Il Sistema ha i meccanismi di  protezione attivi |  |
|  | Il malintenzionato mette in  azione l’attacco |
| Il Sistema notifica il tentativo di DoS  mentre cerca di ridurre l’efficacia  dell’attacco |  |
|  | Il Sistema non riesce a contenere  l’attacco e va fuori uso |
| L’amministratore vede che il  sistema è fuori uso e tenta il  ripristino locale |  |

|  |  |  |
| --- | --- | --- |
| *Titolo* | GarantireProtezione | |
| *Descrizione* | In tutte le trasmissioni i dati devono essere protetti | |
| *Misuse case* | Sniffing, ManInTheMiddle | |
| *Relazioni* | GestioneLettore, GestioneScrittore, GestioneAssistenza,  GestioneAmministratore | |
| *Precondizioni* | 1. L’Attaccante ha i mezzi per intercettare i messaggi nel Sistema 2. L’Attaccante ha i mezzi per modificare i messaggi 3. L’Attaccante ha i mezzi per spedire il messaggio modificato al destinatario | |
| *Postcondizioni* | Il Sistema notifica un tentativo di danno all’integrità dei dati | |
| *Scenario principale* | *Sistema* | *Attaccante* |
| Controlla l’invio e la ricezione dei  messaggi e salva i dati nei log |  |
|  | Intercetta un messaggio inviato |
|  | Non riesce a rimuovere i  meccanismi di sicurezza |
|  | Genera un messaggio e tenta di  inviarlo al destinatario |
| Si accorge che il messaggio non è pertinente e segnala  all’amministratore e all’assistenza  l’avvenimento |  |
| *Scenario di un attacco avvenuto con successo* | *Sistema* | *Attaccante* |
| Controlla l’invio e la ricezione dei  messaggi e salva i dati nei log |  |
|  | Intercetta un messaggio inviato |
|  | Riese a rimuovere i meccanismi  di sicurezza |
|  | Modifica i dati, poi riapplica i  meccanismi di sicurezza e ritrasmette in dati |
| Elabora i dati e procede salvando il  tutto nei log |  |
| Monitora  periodicamente i log alla ricerca di pattern atipici e se  rilevati notifica all’amministratore e  all’assistenza l’avvenimento |  |

##### Requisiti di protezione dei dati

Dall’analisi del rischio si possono evincere i seguenti ulteriori requisiti:

1. Generazione di un log per tenere traccia di
   1. tutte le azioni che si verificano all’interno del sistema
   2. le informazioni scambiate tra le parti del sistema (che vanno protette per evitare che un accesso fraudolento al sistema di log possa rivelare dati riservati)
2. Adottare meccanismi di analisi del log per
   1. Identificare accessi sospetti
   2. Identificare incongruenze tra i messaggi in uscita e in entrata
   3. Individuare una corretta politica di controllo degli accessi
   4. Proteggere i dati memorizzati e trasmessi
3. Il sistema deve essere realizzato in modo da consentire il ripristino locale ad un evento da parte dell’amministratore, nel caso in cui l’attacco DoS sia andato a buon fine.
4. In seguito a tre tentativi di accesso ad un account con credenziali errate, il sistema deve

provvedere a sospendere temporaneamente l’account.

##### Requisiti di sistema aggiornati

|  |  |  |
| --- | --- | --- |
| ID  Requisito | REQUISITO | TIPO |
| R29F | Il sistema, tramite un meccanismo di log, deve memorizzare tutte le operazioni che vengono eseguite sulla piattaforma | Funzionale |
| R30F | Devono essere presenti mezzi per l’analisi del log al fine di  individuare azioni sospette | Funzionale |
| R31F | Il sistema deve avere una struttura tale da permettere il ripristino in seguito ad un attacco DoS | Funzionale |
| R32F | Il sistema può bloccare temporaneamente l’accesso ad un  account | Funzionale |
| R15NF | L’amministratore è l’unico ad avere accesso al log | Non Funzionale |
| R16NF | Il blocco temporaneo dell’account avviene in seguito a 3  tentativi di accesso | Non Funzionale |

#### Glossario aggiornato

|  |  |  |
| --- | --- | --- |
| VOCE | DEFINIZIONE | SINONIMI |
| Log | Registro dove vengono salvate informazioni per risalire ad operazioni critiche svolte. Composto da una  serie di voci | Registro |
| Voce di Log | Si tratta di una riga del log | Record di Log |

#### Casi d’uso aggiornati

Ricerca



GestioneCollezione

GestioneTrial

GestioneCapitolo

Registrazione

<<extend>>

<<extend>>

<<extend>>

<<extend>>

PassaggioAPremium

<<extend>>

GestioneLettore RimozioneTrial

Lettore

Login

<<includ e>>

VisualizzaLog

<<event>>

ScadenzaTrial

<<includ e>>

<<includ e>>

<<extend>>

GestioneScrittore

<<includ e>>

Gestione Amministratore

Scrittore

<<extend>>

Inizio

Sponsorizzazione

Amministratore

FineSponsorizzazione

Gestione

Assistenza

<<event>>

Scadenza Sponsorizzazione

Assistenza

|  |  |
| --- | --- |
| Titolo | VisualizzaLog |
| Descrizione | Permette di visualizzare i log del Sistema |
| Attori | Amministratore |
| Relazioni | GestioneAmministratore |
| Precondizioni | L’Amministratore ha eseguito l’acceso con le proprie credenziali |
| Postcondizioni |  |
| Scenario principale | 1. Il Sistema offre all’Amministratore la possibilità di visualizzare   i log del sistema   1. L’Amministratore può tornare alla schermata principale |
| Scenari alternativi |  |
| Requisiti non funzionali | R15NF |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | Login |
| Descrizione | Permette di accedere al sistema |
| Attori | Amministratore, Assistenza, Scrittore, Lettore |
| Relazioni | GestioneAmministratore, GestioneAssistenza, GestioneScrittore,  GestioneLettore |
| Precondizioni | L’attore deve essere registrato nel sistema |
| Postcondizioni | L’attore è autenticato nel sistema |
| Scenario principale | 1. Il sistema presenta all’attore una maschera per l’inserimento   delle credenziali di accesso: username e password   1. L’attore inserisce i dati del proprio account e conferma 2. Il sistema verifica le credenziali 3. Dopo l’autenticazione viene mostrata la schermata consona al   tipo di attore |
| Scenari alternativi | 1. Credenziali inserite non riconosciute:    1. Il sistema notifica l’attore con un avviso consono    2. Il sistema ripresenta all’attore la maschera di inserimento credenziali 2. Account bloccato:    1. Il sistema notifica l’attore con un avviso consono    2. Il sistema ripresenta all’attore la maschera di inserimento credenziali 3. Credenziali errate inserite troppe volte:    1. Il sistema notifica l’attore con un avviso consono    2. Il sistema blocca l’account dell’utente    3. Il sistema ripresenta all’attore la maschera di inserimento   credenziali |
| Requisiti non funzionali | R5NF, R16NF |
| Punti aperti |  |

# Analisi del problema

## Analisi Documento dei Requisiti

##### Analisi delle Funzionalità

Tabella Funzionalità

I colori sono significativi: i campi con sfondi più scuri rappresentano funzionalità semplici o complesse che si possono ritrovare identiche nel diagramma dei casi d’uso, in caso di funzionalità complesse ci sono di seguito dei campi di colori più chiari che vanno a specificare quali sono le sotto funzionalità che le compongono.

|  |  |  |  |
| --- | --- | --- | --- |
| Funzionalità | Tipo | Grado Complessità | Requisiti Collegati |
| VisualizzaLog | Interazione con l’esterno | Semplice | R29F, R30F, R15NF |
| GestioneAmministratore | Memorizzazione dati, gestione dati, interazione  con l’esterno | Complessa | R24F |
| CreazioneAccountAssistenza | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R24F |
| RimozioneAccountAssistenza | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R24F |
| GestioneAssistenza | Memorizzazione dati, gestione dati, interazione con l’esterno | Complessa | R25F, R26F, R27F |
| BloccoUtente | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R25F |
| SbloccoUtente | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R25F |
| RimozioneCommento | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R26F |
| AccettazioneScrittore | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R27F |

|  |  |  |  |
| --- | --- | --- | --- |
| GestioneLettore | Memorizzazione dati, gestione dati, interazione con l’esterno | Complessa | R7F, R12F, R13F, R14F, R15F, R16F, R17F, R18F, R19F, R20F, R21F, R22F, R23F, R13NF, R14NF |
| Ricerca | Interazione con l’esterno,  gestione dati | Semplice | R13F |
| PassaggioAPremium | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R7F, R22F, R14NF |
| GestioneCapitolo | Memorizzazione dati, gestione dati, interazione  con l’esterno | Complessa | R15F, R16F, R17F, R18F, R19F |
| SelezionaCapitoloComeLetto | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R15F |
| ValutazioneCapitolo | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R16F |
| AssegnazioneEmojiCapitolo | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R17F |
| CommentoCapitolo | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R18F |
| SegnalazioneCommento | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R19F |
| GestioneCollezione |  | Complessa | R14F |
| AggiuntaACollezione | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R14F |
| RimozioneDaCollezione | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R14F |
| GestioneTrial |  | Complessa | R20F, R21F |
| AggiuntaTrial | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R21F |
| LetturaTrial | Interazione con l’esterno | Semplice | R20F |

|  |  |  |  |
| --- | --- | --- | --- |
| GestioneScrittore | Memorizzazione dati, gestione dati, interazione con l’esterno | Complessa | R23F, R7F, R8F, R9F, R10F, R8NF |
| CaricamentoStruttura | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R8F |
| CaricamentoTestoIntegrale | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R8F, R9F |
| Sponsorizzazione | Memorizzazione dati, gestione dati, interazione  con l’esterno | Semplice | R7F, R10F |
| Login | Interazione con l’esterno,  gestione dati | Semplice | R5F, R6F, R5NF |
| Registrazione | Memorizzazione dati | Semplice | R1F, R2F, R3F, R4F, R5F, R3NF, R5NF |
| RimozioneTrial | Memorizzazione dati, gestione dati | Semplice | R28F, R11NF, R12NF |
| FineSponsorizzazione | Memorizzazione dati, gestione dati | Semplice | R11F |

GestioneAmministratore: Tabella Informazioni/Flusso CreazioneAccountAssistenza

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| Username | Semplice | Alto | Input | Non più di 32 caratteri |
| Password | Semplice | Altissimo | Input | Non più di 32 caratteri |

LetturaLog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| Record di lettura Composto da: Timestamp Descrizione  Livello di importanza | Composto  Semplice Semplice Semplice | Alto  Medio Alto Alto | Output  Output Output Output | Esattamente 23 caratteri Non più di 256 caratteri  Enumeratore: Basso, Medio, Alto |

GestioneAssistenza: Tabella Informazioni/Flusso AccettazioneScrittore

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Nome | Semplice | Medio | Input | Non più di 64 caratteri |
| Cognome | Semplice | Medio | Input | Non più di 64 caratteri |
| Documento di identità | Semplice | Altissimo | Input | Il Documento deve essere valido e non scaduto |
| Username Assistente | Semplice | Alto | Input | Non più di 32 caratteri |

BloccoUtente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Username Assistente | Semplice | Alto | Input | Non più di 32 caratteri |

GestioneLettore: Tabella Informazioni/Flusso AssegnazioneEmojiCapitolo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| IdCapitolo Composto da: Username scrittore Titolo libro  Titolo capitolo | Composto  Semplice Semplice Semplice | Basso  Basso Basso Basso | Input  Input Input Input | Non più di 256 caratteri  Non più di 32 caratteri Non più di 112 caratteri Non più di 112 caratteri |
| Emoji | Semplice | Basso | Input |  |
| Username Lettore | Semplice | Basso | Input | Non più di 32 caratteri |

LetturaTrial

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| IdLibro Composto da:  Username scrittore  Titolo | Composto  Semplice Semplice | Basso  Basso Basso | Input  Input Input | Non più di 144 caratteri  Non più di 32 caratteri Non più di 112 caratteri |
| Username Lettore | Semplice | Basso | Input | Non più di 32 caratteri |
| Testo Integrale | Semplice | Alto | Output | Deve essere un file .xml o .pdf |

RimozioneDaCollezione

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| IdLibro Composto da:  Username scrittore  Titolo | Composto  Semplice Semplice | Basso  Basso Basso | Input  Input Input | Non più di 144 caratteri  Non più di 32 caratteri Non più di 112 caratteri |
| Username Lettore | Semplice | Basso | Input | Non più di 32 caratteri |

SegnalazioneCommento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| Username segnalante | Semplice | Basso | Input | Non più di 32 caratteri |
| IdCommento Composto da: IdCapitolo Username lettore  Timestamp | Composto  Composto Semplice  Semplice | Basso  Basso Basso  Basso | Input  Input Input  Input | Non più di 311 caratteri  Non più di 256 caratteri Non più di 32 caratteri  Esattamente 23 caratteri |
| Motivazione segnalazione | Semplice | Basso | Input | Non più di 256 caratteri |

GestioneScrittore: Tabella Informazioni/Flusso CaricamentoStruttura

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| Username Scrittore | Semplice | Basso | Input | Non più di 32 caratteri |
| Struttura del libro Composta da: Titolo del libro  Capitoli | Composto  Semplice Composto | Basso  Basso Basso | Input  Input Input | Non più di 128 caratteri Non meno di 1 capitolo |

Registrazione: Tabella Informazioni/Flusso Caso relativo al lettore

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| Nome | Semplice | Medio | Input | Non più di 32 caratteri |
| Cognome | Semplice | Medio | Input | Non più di 32 caratteri |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Password | Semplice | Alto | Input | Non più di 32 caratteri |

Caso relativo allo scrittore

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| Nome | Semplice | Medio | Input | Non più di 32 caratteri |
| Cognome | Semplice | Medio | Input | Non più di 32 caratteri |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Password | Semplice | Alto | Input | Non più di 32 caratteri |
| Documento d’identità | Semplice | Altissimo | Input | Dimensione massima file di 4MB |

Login: Tabella Informazioni/Flusso

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Password | Semplice | Alto | Input | Non più di 32 caratteri |

FineSponsorizzazione: Tabella Informazioni/Flusso

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| IdLibro Composto da:  Username scrittore  Titolo | Composto  Semplice Semplice | Basso  Basso Basso | Input  Input Input | Non più di 152 caratteri  Non più di 32 caratteri Non più di 120 caratteri |

RimozioneTrial: Tabella Informazioni/Flusso

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Informazione | Tipo | Livello protezione / privacy | Input/output | Vincoli |
| IdLibro Composto da:  Username scrittore  Titolo | Composto  Semplice Semplice | Basso  Basso Basso | Input  Input Input | Non più di 144 caratteri  Non più di 32 caratteri Non più di 112 caratteri |
| Username Utente | Semplice | Basso | Input | Non più di 32 caratteri |

##### Analisi dei Vincoli

Tabella Vincoli

|  |  |  |  |
| --- | --- | --- | --- |
| Requisito | Categorie | Impatto | Funzionalità |
| Velocità di ricerca del libro | Tempo di risposta | Cercare di migliorare | GestioneLettore |
| Semplicità di navigazione  nell’applicazione | Tempo di risposta, usabilità | Cercare di migliorare | GestioneLettore, GestioneScrittore |
| Velocità di registrazione utente | Tempo di risposta | Cercare di migliorare | GestioneLettore, GestioneScrittore |
| Velocità di inserimento del libro con tutti i dati  caratterizzanti | Tempo di risposta | Cercare di migliorare | ScrittoreUsage |
| Protezione di dati sensibili degli utenti per mezzo di meccanismi di  cifratura | Sicurezza | Peggiorano tempo di risposta, migliora la privacy dei dati | GestioneLettore, GestioneScrittore, GestioneAssistenza,  GestioneAmministratore |
| Periodico backup dei dati | Sicurezza, tempo di risposta, usabilità | Peggiorano tempo di risposta, migliorare la privacy e persistenza dei dati | GestioneLettore, GestioneScrittore, GestioneAssistenza, GestioneAmministratore |

##### Analisi delle Interazioni

Tabella Maschere

|  |  |  |
| --- | --- | --- |
| Maschera | Informazioni | Funzionalità |
| View Registrazione | Nome, Cognome, Username, Password,  Tipo di account (lettore/scrittore), documento (solo se il tipo di account è “scrittore”) | GestioneLettore, GestioneScrittore |
| View Login | Username e Password | GestioneLettore, GestioneScrittore, GestioneAssistenza,  GestioneAmministratore |
| Home GestioneLettore | Lista delle letture in corso, Stato degli slot, possibilità di  navigazione verso View Scheda Libro,  View RicercaLibri e View Passaggio a Premium | GestioneLettore |
| View RicercaLibri | Input per la Ricerca, lista di eventuali risultati, possibilità di navigazione verso View Scheda Libro | Ricerca |
| View Scheda Libro | Struttura del Libro, Nome e Cognome dello Scrittore, elenco dei Capitoli con relativi campi di valutazione e Commenti, possibilità di  navigazione verso View Lettura e View Segnalazione | GestioneCollezione (AggiungiACollezione, RimuoviDaCollezione), GestioneTrial (AggiuntaTrial), GestioneCapitolo (Valutazione Capitolo, Assegnazione Emoji Capitolo, Commento Capitolo,  Segnalazione Commento) |
| View Lettura | Testo Integrale del Libro | GestioneTrial (LetturaTrial) |
| View Passaggio A Premium | Nome, Cognome, numero carta, CVV, scadenza carta, circuito/tipo carta | PassaggioAPremium |
| View Sponsorizzazione | IdLibro, Nome, Cognome, numero  carta, CVV, scadenza carta, circuito/tipo carta | GestioneScrittore (Sponsorizzazione) |
| View Segnalazione | Input per la specifica del motivo della Segnalazione del Commento | GestioneCapitolo (SegnalazioneCommento) |
| Home GestioneScrittore | Lista dei propri libri, possibilità di navigazione verso Home | GestioneScrittore |

|  |  |  |
| --- | --- | --- |
|  | GestioneLettore, View Sponsorizzazione, View Inserimento Libro |  |
| View Inserimento Libro | Sezione per libro già presente: IdLibro, Testo Integrale Sezione per libro non presente:  Titolo del libro, Username Scrittore,  Titoli dei capitoli, eventuale Testo Integrale | GestioneScrittore (CaricamentoStruttura, CaricamentoTestoIntegrale) |
| Home GestioneAssistenza | Lista degli utenti bloccati, Lista segnalazioni, possibilità di navigazione verso View Accettazione Scrittore, View Gestione Ban e View Rimozione Commento | GestioneAssistenza |
| View Accettazione Scrittore | Username, Nome Scrittore, Cognome Scrittore, Documento d’identità | GestioneAssistenza (AccettazioneScrittore) |
| View Gestione Ban | Username Utente | GestioneAssistenza (BloccoUtente, SbloccoUtente) |
| View Rimozione Commento | Commento | GestioneCapitolo (RimozioneCommento) |
| Home GestioneAmministratore | Lista degli account Assistenza attivi, possibilità di  navigazione verso View Controllo Assistenza, View Visualizzazione Log  ed Home GestioneAssistenza | GestioneAmministratore |
| View Controllo Assistenza | Username, Password | GestioneAmministratore (Creazione Account Assistenza,  Rimozione Account Assistenza) |
| View Visualizzazione Log | Timestamp, Descrizione, Livello | VisualizzaLog |

Tabella Sistemi Esterni

|  |  |  |  |
| --- | --- | --- | --- |
| Sistema | Descrizione | Protocollo di Interazione | Livello di Sicurezza |
| PayPal | Servizio che consente di pagare, inviare denaro e accettare pagamenti | PayPal mette a disposizione delle funzionalità che permettono di eseguire transazioni tramite vari tipi di  carte | Alto livello di sicurezza perché si devono gestire spostamenti di denaro |

## Analisi Ruoli e Responsabilità

Tabella Ruoli

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ruolo | Responsabilità | Maschere | Riservatezza | Numerosità |
| Lettore | Garantire la privacy della password impostata | View Registrazione, View Login,  Home GestioneLettore, View Scheda Libro, View Lettura,  View Passaggio A Premium, View Segnalazione,  View Ricerca Libri | È richiesto un basso/medio grado di riservatezza | Il numero può essere gestito da un amministratore di sistema,  al più e limitato dalle risorse del sistema |
| Scrittore | Garantire la privacy della password impostata | View Registrazione, View Login,  Home GestioneLettore, View Scheda Libro, View Lettura,  View Passaggio A Premium, View Segnalazione,  View Ricerca Libri,  Home GestioneScrittore, View Inserimento Libro,  View Sponsorizzazione | È richiesto un medio grado di riservatezza | Il numero può essere gestito da un amministratore di sistema,  al più è limitato dalle risorse del sistema |
| Assistenza | Visualizzazione di tutti gli Utenti e dei loro commenti | View Login,  Home GestioneAssistenza, View  Accettazione Scrittore, View Gestione Ban,  View Rimozione Commento | È richiesto un alto grado di riservatezza | Un team costituito da un numero considerevole di persone scelte dall’Amministratore |
| Amministratore | Gestione di tutte  le informazioni relative agli utenti ed agli assistenti e al sistema | View Login,  Home GestioneAssistenza, View  Accettazione Scrittore, View Gestione Ban, View Rimozione Commento,  Home GestioneAmministratore, View Controllo Assistenza,  View Visualizzazione Log | È richiesto un alto grado di riservatezza | Un’unica persona |

Lettore: Tabella Ruolo-Informazioni

|  |  |
| --- | --- |
| Informazione | Tipo di Accesso |
| Nome Lettore | Lettura/scrittura |
| Cognome Lettore | Lettura/scrittura |
| Username | Lettura/scrittura |
| Password | Scrittura |
| Struttura del Libro | Lettura |
| Nome Scrittore | Lettura |
| Cognome Scrittore | Lettura |
| Testo Integrale | Lettura |
| Commento | Lettura/scrittura |
| Emoji | Lettura/scrittura |
| Messaggio segnalazione | Scrittura |
| Timestamp segnalazione | Scrittura |
| Valutazione | Lettura/scrittura |
| Transazione | Scrittura |

Scrittore: Tabella Ruolo-Informazioni

|  |  |
| --- | --- |
| Informazione | Tipo di Accesso |
| Nome Scrittore | Lettura/scrittura |
| Cognome Scrittore | Lettura/scrittura |
| Username | Lettura/scrittura |
| Password | Scrittura |
| Documento | Scrittura |
| Struttura del Libro | Lettura/scrittura |
| Testo Integrale | Lettura/scrittura |
| Commento | Lettura/scrittura |
| Emoji | Lettura/scrittura |
| Messaggio segnalazione | Scrittura |
| Timestamp segnalazione | Scrittura |
| Valutazione | Lettura/scrittura |
| Transazione | Scrittura |

Assistenza: Tabella Ruolo-Informazioni

|  |  |
| --- | --- |
| Informazione | Tipo di Accesso |
| Username | Lettura/scrittura |
| Password | Scrittura |
| Nome Utente | Lettura |
| Cognome Utente | Lettura |
| Username Utente | Lettura |
| Documento Scrittore | Lettura |
| Messaggio Segnalazione | Lettura |
| Timestamp Segnalazione | Lettura |

Amministratore: Tabella Ruolo-Informazioni

|  |  |
| --- | --- |
| Informazione | Tipo di Accesso |
| Username Amministratore | Scrittura |
| Password Amministratore | Scrittura |
| Username Assistente | Lettura/scrittura |
| Password Assistente | Scrittura |
| Nome Utente | Lettura |
| Cognome Utente | Lettura |
| Username Utente | Lettura |
| Documento Scrittore | Lettura |
| Messaggio Segnalazione | Lettura |
| Timestamp Segnalazione | Lettura |
| Voce di Log | Lettura |

## Scomposizione del Problema

Tabella Scomposizione Funzionalità

|  |  |
| --- | --- |
| Funzionalità | Scomposizione |
| GestioneLettore | GestioneCollezione, GestioneTrial, GestioneCapitolo, Ricerca,  PassaggioAPremium |
| GestioneCollezione | AggiuntaACollezione, RimozioneDaCollezione |
| GestioneTrial | AggiuntaTrial LetturaTrial |
| GestioneCapitolo | SelezionaCapitoloComeLetto, ValutazioneCapitolo, CommentoCapitolo, AssegnazioneEmojiCapitolo,  SegnalazioneCommento |
| GestioneScrittore | CaricamentoStruttura, CaricamentoTestoIntegrale, Sponsorizzazione |
| GestioneAssistenza | AccettazioneScrittore, BloccoUtente, SbloccoUtente,  RimozioneCommento |
| GestioneAmministratore | CreazioneAccountAssistenza, RimozioneAccountAssistenza |

Tabella Sotto-Funzionalità: GestioneCollezione

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-funzionalità | Sotto-funzionalità | Legame | Informazioni |
| RimozioneDaCollezione | AggiuntaACollezione | Non si può rimuovere un Libro dalla Collezione se non è stato aggiunto in  precedenza | Username Utente, IdLibro |

Tabella Sotto-Funzionalità: GestioneTrial

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-funzionalità | Sotto-funzionalità | Legame | Informazioni |
| LetturaTrial | AggiuntaTrial | Non si può leggere un Trial se non è stato aggiunto in precedenza ad uno  degli Slot disponibili | Username Utente, IdLibro |

Tabella Sotto-Funzionalità: GestioneCapitolo

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-funzionalità | Sotto-funzionalità | Legame | Informazioni |
| ValutazioneCapitolo | SelezioneCapitolo ComeLetto | Non si può valutare un capitolo che non sia stato già selezionato come  letto | Associazione tra utente e capitolo letto |
| Assegnazione EmojiCapitolo | SelezioneCapitolo ComeLetto | Non si può assegnare un’emoji ad un capitolo che non sia stato già selezionato  come letto | Associazione tra utente e capitolo letto |
| CommentoCapitolo | SelezioneCapitolo ComeLetto | Non si può commentare un capitolo che non sia stato già selezionato  come letto | Associazione tra utente e capitolo letto |
| SegnalazioneCommento | CommentoCapitolo | Non si può segnalare un commento che non sia stato già  rilasciato | IdCommento |

Tabella Sotto-Funzionalità: GestioneScrittore

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-funzionalità | Sotto-funzionalità | Legame | Informazioni |
| CaricamentoTestoIntegrale | CaricamentoStruttura | Non si può aggiungere il testo integrale di un libro di cui non è presente la  struttura | Username Scrittore, Titolo Libro |
| Sponsorizzazione | CaricamentoStruttura | Non si può sponsorizzare un Libro che non sia  stato già caricato | Username Scrittore, Titolo Libro |

Tabella Sotto-Funzionalità: GestioneAssistenza

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-funzionalità | Sotto-funzionalità | Legame | Informazioni |
| SbloccoUtente | BloccoUtente | Non si può sbloccare un Utente se prima non è stato bloccato | Username Utente |
| BloccoUtente | AccettazioneScrittore | Non si può bloccare uno scrittore che non è ancora stato  accettato | Username Utente |

Tabella Sotto-Funzionalità: GestioneAmministratore

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-funzionalità | Sotto-funzionalità | Legame | Informazioni |
| RimozioneAccountAssistenza | CreazioneAccountAssistenza | Non si può rimuovere un account se prima non è  stato creato | Username e password dell’Assistenza |

## Modello del dominio

Di seguito è riportato il diagramma delle classi del modello del dominio in relazione alla gestione del capitolo: mostra quindi le relazioni tra lettore e capitolo.

assegna

1

1

\*

\*

1

\*

\*

\*

Segnalazione

-nome: String

-cognome: String

+selezionaCapitoloComeLetto()

+aggiungiACollezione()

+rimuoviDaCollezione()

+aggiungiTrial()

\*

\*

\*

\*

\*

applica

Emoji

-id: String

-id: String

-timestamp: Date

-motivazione: String

Lettore

+segnalaCommento()

+valutaCapitolo()

+assegnaEmojiCapitolo()

+commentaCapitolo()

-id: String

-timestamp: Date

-messaggio: String

-id: String

-titolo: String

Commento

Capitolo

scrive

1

-id: String

-stelle: Integer

Valutazione

Di seguito è riportato il diagramma delle classi del modello in relazione agli account.

Nel diagramma è presente una relazione di *overlapped* tra tutte le classi che ereditano da Utente: quindi tra le classi di Lettore (e quindi anche Scrittore) con UtenteBloccato/UtenteAttivo e UtentePremium/UtenteStandard. In questo modo sulle classi di Lettore e Scrittore potranno essere invocati i metodi delle altre classi che ereditano da Utente.

|  |
| --- |
| Account |
| -username: String |
|  |

|  |
| --- |
| Amministratore |
|  |
|  |

Di seguito è riportato il diagramma delle classi del dominio relativo alla gestione dei libri: in particolare le relazioni tra lettori o scrittori con i libri.



complete, disjoint

complete, disjoint

complete,disjoint

Lettore

+caricaStruttura()

-documento: File

Scrittore

-nome: String

-cognome: String

Assistente

Utente

+bloccaUtente()

+sbloccaUtente()

+selezionaCapitoloComeLetto()

+aggiungiACollezione()

+rimuoviDaCollezione()

+aggiungiTrial()

UtenteAttivo

UtenteBloccato

+passaAPremium()

UtenteStandard

UtentePremium

|  |  |
| --- | --- |
| Scrittore | |
| -documento: File | |
| +caricaStruttura() | |
| \*  1 |  |
| Libro | |
| -id: String  -titolo: String | |
|  | |

\*



\*

Capitolo

1..n 1

\*

letturaInCorso

\*

complete, disjoint

complete, disjoint

0..3

1

LibroInTrial

LibroNonInTrial

LibroSponsorizzato

LibroNonSponsorizzato

1

-testoIntegrale: File

+leggi()

+aggiungiTestoIntegrale()

+sponsorizza()

Transazione

Slot

-inizio: Date

-id: String

-timestamp: DateTime

sponsorizzato tramite

complete, disjoint

0..1

1

effettua

1

+passaAPremium()

UtenteStandard

Sponsorizzazione

PassaggioAPremium

+selezionaCapitoloComeLetto()

+aggiungiACollezione()

+rimuoviDaCollezione()

+aggiungiTrial()

-nome: String

-cognome: String

Lettore

-id: String

-titolo: String

Di seguito è presente il diagramma delle classi del modello del dominio in relazione alla gestione dei Log.

|  |  |
| --- | --- |
| <<Enumeration>>  Livello | |
| +Basso  +Medio  +Alto | |
| \*  1 |  |
| Voce di Log | |
| -timestamp: DateTime  -descrizione: String | |
|  | |
|  |  |
| Log | |
|  | |
| +visualizzaLog()  +aggiungiLog(VoceDiLog) | |

## Architettura logica

##### Struttura

*Diagramma dei package*



uses

uses

uses uses

uses

uses

uses

uses uses

uses

uses

uses

uses

uses

uses

uses

uses

uses

uses

uses

uses

uses

uses

Diagramma delle classi: Dominio

Non viene riportato il diagramma delle classi associato al package Dominio in quanto è il modello del dominio creato nella fase precedente.

*Diagramma delle classi: Registrazione*



uses

+registraUtente(Nome, Cognome, Username, Password)

+registraUtente(Nome, Cognome, Username, Password, DocumentoDiIdentità)

Registrazione Controller

+aggiungiLog(VoceDiLog)

Logger

ViewRegistrazione

*Diagramma delle classi: Login*



uses

+aggiungiLog(VoceDiLog)

uses

+verificaCredenziali(Username, Password)

Login Controller

Logger

ViewLogin

*Diagramma delle classi: Amministratore e Assistenza*

51



Home Gestione Amministratore

Home GestioneAssistenza

View Visualizzazione Log

View Controllo Assistenza

View GestioneBan

View RimozioneCommento

View AccettazioneScrittore

<<uses>>

<<uses>>

Logger

+aggiungiLog(VoceDiLog)

<<uses>>

GestioneAmministratore Controller

+getAssistenti()

+creaAccountAssistenza(Username,

Password)

+rimuoviAccountAssistenza(Username)

<<uses>>

<<uses>>

<<uses>>

GestioneAssistenza Controller

1

1

Log Controller

<<uses>>

+getVociDiLog(DateTime, DateTime)

+getVociDiLog(Date)

+getScrittoriDisabilitati()

+getUtentiBloccati()

+getSegnalazioni()

+bloccaUtente(Username)

+sbloccaUtente(Username)

+rimuoviCommento(IdCommento)

+accettaScrittore(Username)

<<uses>>

*Diagramma delle classi: Scrittore e Lettore*

52



<<uses>>

<<uses>>

<<uses>>

<<uses>>

<<uses>>

1

1

1

1 1

1

1

1

1

<<uses>>

1

1

1

1

<<uses>>

<<uses>>

<<uses>>

<<uses>>

<<uses>>

+visualizzaTesto(Username, IdLibro)

+segnalaCommento(Username,

IdCommento, Motivazione)

Lettura Controller

Segnalazione Controller

+sponsorizza(Username, IdLibro)

+inserisciLibro(Username,Titolo,Capitoli)

+inserisciLibro(Username,Titolo,Capitoli,

Testo)

+aggiungiTestoIntegrale(Username,

IdLibro,Testo)

+getCommenti(IdCapitolo)

+aggiungiACollezione(Username, IdLibro)

+rimuoviDaCollezione(Username, IdLibro)

+aggiungiTrial(Username, IdLibro)

+selezionaCapitoloComeLetto(Username,

IdCapitolo )

+commentaCapitolo(Username,

IdCapitolo , Commento)

+valutaCapitolo(Username, IdCapitolo,

Valutazione)

+reagisciACapitolo(Username, IdCapitolo,

Emoji)

Sponsorizzazione Controller

InserimentoLibro Controller

SchedaLibro Controller

<<uses>>

1

+passaAPremium(Username)

+getUsername(): Username

+getLibri(Titolo, Tipo)

+aggiungiLog(VoceDiLog)

+getUsername(): Username

PassaggioAPremium Controller

GestioneLettore Controller

RicercaLibro Controller

Logger

GestioneScrittore Controller

<<uses>>

<<uses>>

View Lettura

View Segnalazione

View Sponsorizzazione

View InserimentoLibro

View PassaggioAPremium

View SchedaLibro

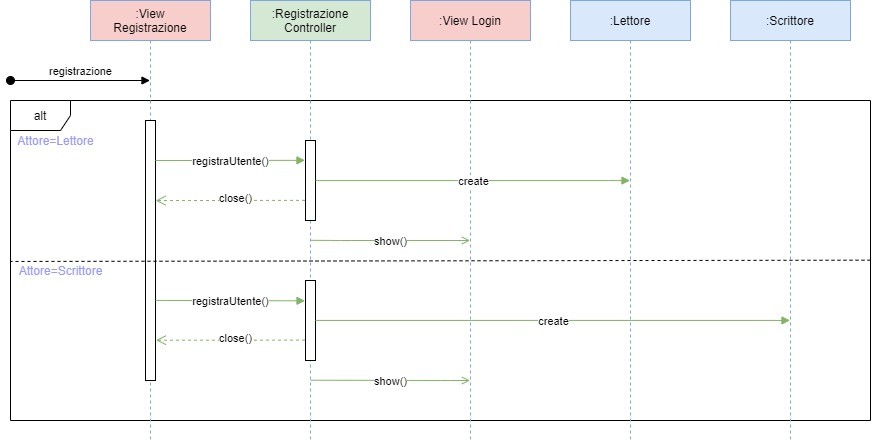
View RicercaLibri

Home GestioneScrittore

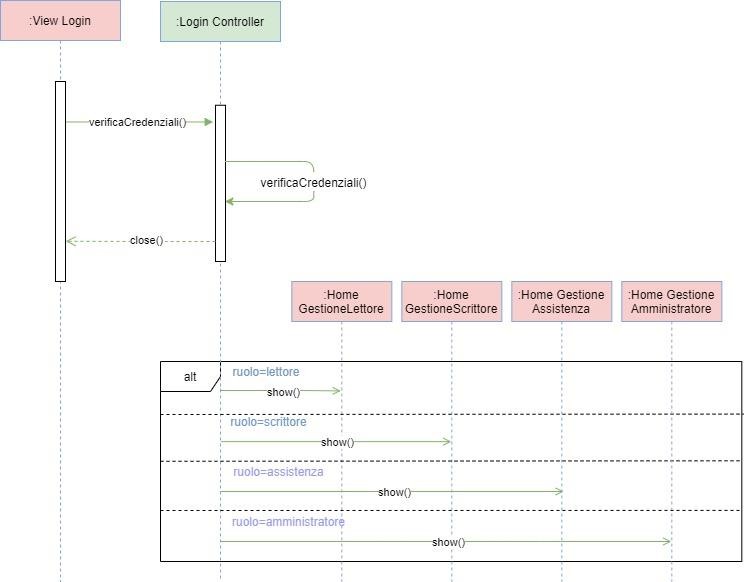
Home GestioneLettore

##### Interazione

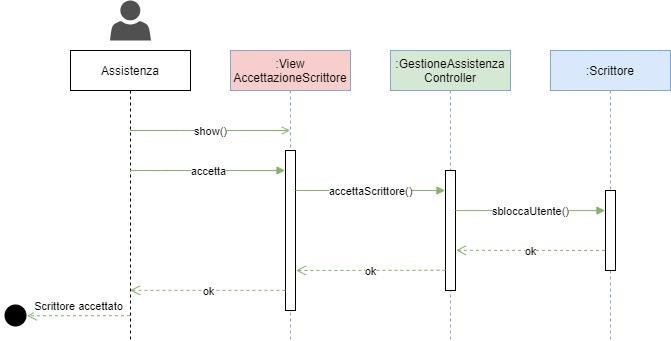
*Diagramma di sequenza: Registrazione*



*Diagramma di sequenza: Login*

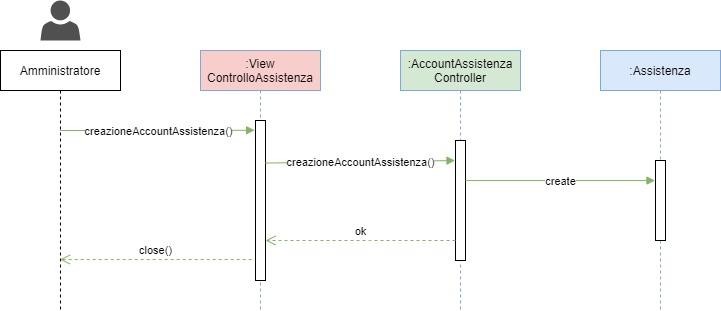


*Diagramma di sequenza: Accettazione Scrittore*

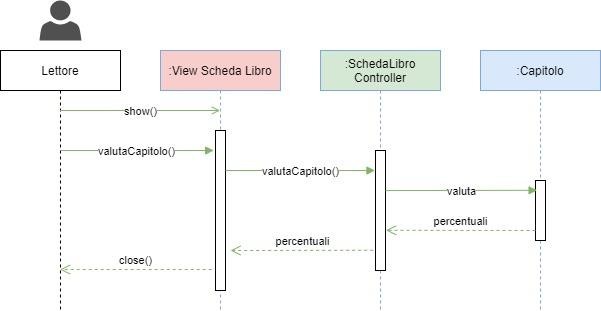


In virtù della relazione non disgiunta tra Scrittore e UtenteBloccato è possibile invocare il metodo sbloccaUtente() su Scrittore.

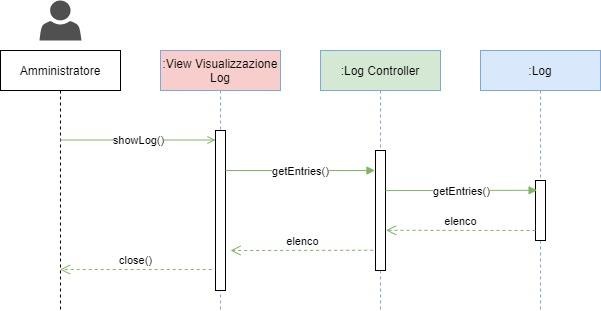
*Diagramma di sequenza: Creazione Account Assistenza*



*Diagramma di sequenza: Valutazione Capitolo*

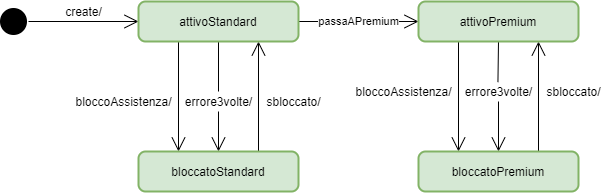


*Diagramma di sequenza: Visualizzazione Log*

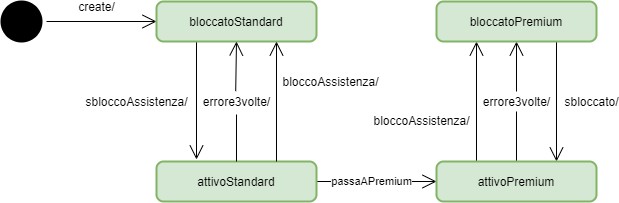


##### Comportamento

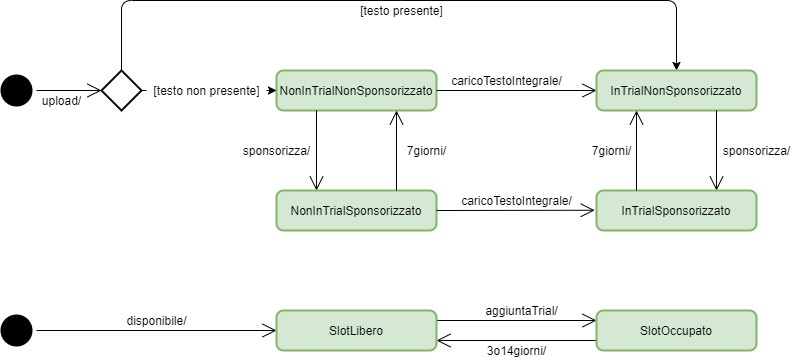
*Diagramma di Stato: Lettore*



*Diagramma di Stato: Scrittore*



*Diagramma di Stato: Libro e Slot*



## Piano di Lavoro

Il progetto e lo sviluppo del sistema sono assegnati a diversi team come indicato nella tabella sottostante:

|  |  |
| --- | --- |
| Nome Team | Composizione |
| Team progettazione | Cialone, Imboccioli, Marinelli |
| Team sviluppo | Cialone, Imboccioli, Marinelli |
| Team DB | Cialone, Imboccioli, Marinelli |
| Team sicurezza | Cialone, Imboccioli, Marinelli |
| Team grafico | Cialone, Imboccioli, Marinelli |

|  |  |  |
| --- | --- | --- |
| Package | Progetto | Sviluppo |
| Dominio | Team progettazione + Team  DB | Team sviluppo + Team DB |
| GestioneLettore | Team progettazione | Team sviluppo |
| GestioneScrittore | Team progettazione | Team sviluppo |
| GestioneAssistenza | Team progettazione | Team sviluppo |
| GestioneAmministratore | Team progettazione | Team sviluppo |
| Registrazione | Team sicurezza | Team sviluppo sicurezza |
| Login | Team sicurezza | Team sviluppo sicurezza |
| Log | Team sicurezza | Team sviluppo sicurezza |
| InterfacciaLettore | Team progettazione + Team  grafico | Team sviluppo + Team  Grafico |
| InterfacciaScrittore | Team progettazione + Team  grafico | Team sviluppo + Team  Grafico |
| InterfacciaAssistenza | Team progettazione + Team  grafico | Team sviluppo + Team  grafico |
| InterfacciaAmministratore | Team progettazione + Team  grafico | Team sviluppo + Team  grafico |
| InterfacciaRegistrazione | Team progettazione + Team  grafico | Team sicurezza + Team  grafico |
| InterfacciaLogin | Team sicurezza + Team  grafico | Team sicurezza + Team  grafico |
| InterfacciaLog | Team sicurezza + Team  grafico | Team sicurezza + Team  grafico |
| GestionePersonale | Team progettazione + Team  DB | Team sviluppo + Team DB |

I tempi di rilascio previsti sono i seguenti:

* Progettazione entro 4 settimane dalla data odierna
* Sviluppo delle singole parti con collaudo unitario entro 5 settimane rispetto al fine della progettazione
* Integrazione e test dell’intero sistema entro 3 settimane rispetto alla fine dello sviluppo

##### Sviluppi futuri

Il committente ha richiesto che nei prossimi mesi si provveda anche ad aggiungere nuove emoji e nuove opzioni di valutazione associati ai capitoli.

Si deve, inoltre, mantenere aperta la possibilità di avere nuovi ruoli associati agli utenti. In relazione ai libri, nel caso in cui vengano inserite nuove forme di fruizione di opere, queste devono poter essere aggiunte agevolmente.

Si richiede al Team progettazione di tenere conto di questi sviluppi futuri.

## Piano del Collaudo

Per garantire il corretto funzionamento del sistema sono necessari una gamma di test unitari e di integrazione che permettano la correttezza delle singole parti. Di seguito vengono riportati i casi più di interesse.

*//package test.dominio;*

import static org.junit.Assert.\*; import java.util.ArrayList;

public class **TestLettore** { private Lettore lettore;

@Before

public void **SetUp**() {

lettore = new **Lettore**("MarioRed96", "Mario", "Rossi");

}

@Test

public void **TestGetterLettore**() { **assertEquals**(lettore.**getUsername**(), "MarioRed96"); **assertEquals**(lettore.**getName**(), "Mario"); **assertEquals**(lettore.**getCognome**(), "Rossi");

**assertEquals**(lettore.collezione.**getId**(), "MarioRed96@collezione");

**assertEquals**(lettore.trial.**getId**(), "MarioRed96@trial");

}

@Test

public void **TestSetterLettore**() { lettore.**setUsername**("BookLover83"); **assertEquals**(lettore.**getUsername**(), "BookLover83"); lettore.**setNome**("Paolo"); **assertEquals**(lettore.**getName**(), "Paolo"); lettore.**setCognome**("Bianchi"); **assertEquals**(lettore.**getCognome**(), "Bianchi");

}

}

public class **TestScrittore** { private Scrittore scrittore;

@Before

public void **SetUp**() {

scrittore = new **Scrittore**("BigBrother84", "George", "Orwell", documento); capitoli = Array.**asList**("parte1", "parte2", "parte3");

libro = new **Libro**(scrittore, "1984", capitoli);

}

@Test

public void **TestGetterScrittore**() { **assertEquals**(scrittore.**getUsername**(), "BigBrother84"); **assertEquals**(scrittore.**getName**(), "George"); **assertEquals**(scrittore.**getCognome**(), "Orwell");

**assertEquals**(scrittore.collezione.**getId**(), "BigBrother84@collezione"); **assertEquals**(scrittore.trial.**getId**(), "BigBrother84@trial"); **assertNotNull**(scrittore.documento);

}

@Test

public void **TestSetterLibro**() { **assertEquals**(libro.**getScrittore**(), scrittore); **assertEquals**(libro.**getTitolo**(), "1984");

**assertArrayEquals**(libro.**getCapitoli**().**toArray**(), capitoli.**toArray**());

**assertEquals**(libro.**getId**(), "BigBrother84@1984");

}

@Test

public void **TestSetterScrittore**() { scrittore.**setUsername**("DevilDeal54"); **assertEquals**(scrittore.**getUsername**(), "DevilDeal54"); scrittore.**setNome**("Oscar"); **assertEquals**(scrittore.**getName**(), "Oscar"); scrittore.**setCognome**("Wilde"); **assertEquals**(scrittore.**getCognome**(), "Wilde");

}

}

# Progettazione

## Progettazione Architetturale

##### Requisiti non funzionali

Nell’Analisi del Problema (Tabella Vincoli) sono emersi i seguenti requisiti non funzionali che impongono dei vincoli al sistema:

1. Velocità di ricerca e inserimento del libro e di registrazione degli utenti
2. Semplicità di navigabilità dell’applicazione
3. Protezione dei dati sensibili degli utenti per mezzo di meccanismi di cifratura
4. Periodico backup dei dati
5. Controllo degli accessi

Uno degli aspetti chiave dell’applicazione è l’usabilità: la piattaforma deve essere intuitiva e

semplice da utilizzare per soddisfare i requisiti e le aspettative di un’ampia clientela. Allo stesso tempo si vuole che i dati degli utenti vengano tutelati sia in fase di accesso che in fase di registrazione per cui è fondamentale trovare il giusto compromesso tra semplicità e sicurezza dei dati. Ragion per cui si è deciso di utilizzare il protocollo HTTPS a livello applicativo.

Il controllo degli accessi permette all’assistenza e all’amministratore di monitorare la piattaforma al fine di garantirne il lecito utilizzo. In particolare, permette ai medesimi di sfruttare i loro privilegi per far sì che chiunque usufruisca dell’applicazione lo possa fare in modo sicuro.

##### Scelta dell’architettura

Dal punto di vista architetturale, la scelta più adatta è risultata essere l’architettura client/server a

tre livelli rappresentati da database, application server e client.

L1 – Client

Il Client avrà modo di connettersi al server tramite una connessione TLS, il che permetterà di usufruire delle funzionalità presenti nel server mantenendo un livello di sicurezza adeguato.

L2 – Server

Per quanto riguarda il lato Server è stato ritenuto opportuno avere più server specializzati in funzionalità diverse:

* Un server per la gestione dei Log
* Vari server per la logica applicativa
  + Server per le funzionalità dell’Amministratore
  + Server per le funzionalità dell’Assistenza
  + Server per le funzionalità dello Scrittore
  + Server per le funzionalità del Lettore
* Un server per la gestione dei Login
* Un server per la gestione della Registrazione

L3 – Database

Al fine di mantenere i dati persistenti si è scelto l’utilizzo di un server dedicato nel quale sarà installato un opportuno DBMS che gestirà la base dati della piattaforma Bhooked (il cui schema di dati si può trovare nelle pagine successive) ed un ulteriore server sarà utilizzato per la memorizzazione dei log. La separazione tra log e database è fondamentale in modo da evitare che eventuali compromissioni di una macchina alterino anche l’altra. È sempre necessario essere in una condizione che permetta di riportare i dati del sistema ad uno stato consistente ed integro.

La metodologia “forza bruta”, con l’utilizzo di metodi CRUD, permetterà l’interfacciamento con il

DBMS.

In aggiunta si è ritenuto utile il pattern Broker per gestire la sessione e per aggiungere, in questo,

astrazione e protezione. L’utilizzo del broker è particolarmente utile a disaccoppiare il client dai server poiché nasconde questi ultimi. In questo modo risulterà più semplice aggiungere o rimuovere server. Infine, per garantire la sicurezza nella comunicazione tra client e server si è deciso di utilizzare il protocollo TLS che, essendo un protocollo crittografico, permette una comunicazione sicura dalla sorgente al destinatario (end-to-end) su reti TCP/IP fornendo autenticazione, integrità e protezione dei dati, operando al di sopra del livello di trasporto.

Si è deciso di utilizzare il pattern MVP per separare i componenti dall’applicazione.

##### Diagramma dei package

Nella figura sottostante è riportata l’Architettura del Sistema organizzata attraverso un diagramma

dei package.

|  |
| --- |
| InterfacciaRegistrazione |
| InterfacciaLogin |
| InterfacciaLettore |
| InterfacciaScrittore |
| InterfacciaAssistenza |
| InterfacciaAmministratore |

Broker

GestioneLettore

GestioneScrittore

Registrazione

Login

Dominio Lettore

Dominio Scrittore

Dominio Registrazione

Dominio Login

GestioneAmministratore

Gestion Assistenza

Log

Dominio Amministratore

Dominio Assistenza

Dominio Log

##### Diagramma dei componenti

Nella figura sottostante è riportata l’Architettura del Sistema organizzata attraverso un diagramma dei componenti.



richiesta

**Broker**

servizio

**Server**

**Amministratore**

**ServerAssistenza**

**ServerScrittore**

**ServerLettore**

**Server**

**Registrazione**

**ServerLogin**

**Client**

**PayPal**

##### Scelte tecnologiche



**DBMS**



**Log**

Poiché si lavora in [Java](https://it.wikipedia.org/wiki/Java_(linguaggio_di_programmazione)) con diversi tipi di file: [JavaServer Pages](https://it.wikipedia.org/wiki/JavaServer_Pages), [servlet](https://it.wikipedia.org/wiki/Servlet), classi Java, [XML](https://it.wikipedia.org/wiki/XML) ecc. si è deciso di utilizzare per il Deployment l’archivio con [estensione](https://it.wikipedia.org/wiki/Estensione_(informatica)) .war, [acronimo](https://it.wikipedia.org/wiki/Acronimo) di Web

application ARchive. Questa scelta è stata decisa anche tendendo conto del fatto che questo tipo di archivio viene utilizzato dai [programmatori](https://it.wikipedia.org/wiki/Programmatore) Java proprio per distribuire tutto l'applicativo software sviluppato.

## Progettazione di dettaglio

Nel seguito si riportano i diagrammi di dettaglio delle varie parti del Sistema.

##### Struttura

*Diagramma di Dettaglio: Dominio – Interazione con il capitolo*

Capitolo

-titolo: String

+getTitolo(): String

\* +getCommenti(): List<Commento>

+getValutazioni(): List<Valutazione>

+getLibro(): Libro

+getId(): String

+valutaCapitolo(valutazione: Valutazione): void

+assegnaEmojiCapitolo(emoji: Emoji): void

+commentaCapitolo(commento: String): void

\*

-\_commenti

1

-\_capitolo

Commento

-timestamp: LocalDateTime



-messaggio: String

-eliminato: Boolean

+getTimestamp(): LocalDateTime

+getMessaggio(): String

+getCapitolo(): Capitolo

+getUtente(): Utente

+getSegnalazioni(): List<Segnalazione>

+isEliminato(): Boolean

+getId(): String

+segnalaCommento(segnalazione: Segnalazione)

\*

-\_segnalazioni

1

-\_valutazioni \*

-\_capitolo 1

*Valutazione*

-\_utente 1

\*

-\_commento

Segnalazione

-timestamp: LocalDateTime

-motivazione: String

#tipoValutazione: String

+getTipoValutazione(): String

*+getValutazione(): Number*

+getCapitolo(): Capitolo

+getUtente(): Utente

Stelle

-stelle: Integer

-tipoValutazione: String

+getValutazione(): Number

1

-\_utente

*Utente*

-nome: String

-cognome: String

+getNome(): String

\* +setNome(): void \*

+getCognome(): String

+setCognome(): void

+getStatoUtente(): StatoUtente

+setStatoUtente(StatoUtente): void

+getTipoUtente(): TipoUtente

+setTipoUtente(TipoUtente): void

+getTransazioni(): List<Transazione>

-\_utenteSegnalante

+getTimestamp(): LocalDateTime

+getMotivazione(): String

+getCommento(): Commento

+getUtenteSegnalante(): Utente

+getUtenteSegnalato(): Utente

+getId(): String

1

\*

*Emoji*

\*

#nomeEmoji: String

#bitmap: File

#dimensioniOriginali: Double[][]

-emojiIstanziate: Map<nomeEmoji: String, emoji: Emoji>

+getEmoji(nomeEmoji: String): Emoji

+getNomeEmoji(): String

+getBitmap(): File

+getDimensioniOriginali(): Double[][]

+getEmojis(): List<Emoji>

Confuso

Deluso

Appassionato

#nomeEmoji: String

#bitmap: File

#dimensioniOriginali: Double[][]

#nomeEmoji: String

#bitmap: File

#dimensioniOriginali: Double[][]

#nomeEmoji: String

#bitmap: File

#dimensioniOriginali: Double[][]

Annoiato

Incuriosito

#nomeEmoji: String #bitmap: File

#dimensioniOriginali: Double[][]

#nomeEmoji: String #bitmap: File

#dimensioniOriginali: Double[][]

Valutazione: si è pensato di utilizzare una classe astratta per modellare il concetto di valutazione. Valutazione dichiara solamente l'interfaccia che in futuro potrà essere specializzata in altre classi. Per esempio, non valutazione stelle, bensì voti da 0 a 10, rendendo tutto trasparente al capitolo.

Per questo motivo si applica il *Principio di O/C*, in quanto si prevede che in futuro la classe commento non cambierà mai e lo stesso vale per le valutazioni, poiché Capitolo

fa uso di astrazioni immutabili nel tempo.

Emoji: l’ispirazione parte dal *Pattern Flyweight* in quanto si vuole che ogni tipo emoji sia istanziata una sola volta nel sistema per assicurare una condivisione leggera dell’elemento.

Commento: poiché tra le funzionalità è presente la rimozione dei commenti, piuttosto che una rimozione reale dal DB si è preferito inserire un booleano che influenzasse la visibilità del commento in modo da permettere la tracciabilità degli eventi ed eventuali indagini dell’autorità giudiziaria.

*Diagramma di Dettaglio: Dominio – Account*



-\_utente

1

+getNomeStato(): String

+getTimeStampUpdate: LocalDateTime

+getUtente(): Utente

+getUsername(): String

+setUsername(): void

complete, disjoint

complete,disjoint

-\_statoUtente

1

-\_tipoUtente

1

-\_utente 1

+getTipoUtente(): String

+getTimeStampUpdate: LocalDateTime

+getUtente(): Utente

#tipoUtente: String

#TimeStampUpdate: LocalDateTime

*TipoUtente*

#nomeStato: String

#TimeStampUpdate: LocalDateTime

#nomeStato: String

#TimeStampUpdate: LocalDateTime

#nomeStato: String

#TimeStampUpdate: LocalDateTime

UtenteAbilitato

UtenteBloccato

UtenteDisabilitato

+getNome(): String

+setNome(): void

+getCognome(): String

+setCognome(): void

+getStatoUtente(): StatoUtente

+setStatoUtente(StatoUtente): void

+getTipoUtente(): TipoUtente

+setTipoUtente(TipoUtente): void

+getTransazioni(): List<Transazione>

-nome: String

-cognome: String

Assistente

*Utente*

-username: String

*Account*

#nomeStato: String

#TimeStampUpdate: LocalDateTime

*StatoUtente*

|  |
| --- |
| Amministratore |
|  |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | complete,disjoint | | |  |
| UtentePremium | |  | UtenteStandard | |
| #tipoUtente: String  #TimeStampUpdate: LocalDateTime | | #tipoUtente: String  #TimeStampUpdate: LocalDateTime | |
|  | |  | |

|  |  |
| --- | --- |
|  |  |
| Lettore | |
|  | |
| +getLibriInLettura(): List<Libro>  +getLibroInTrial(): LibroInTrial  +selezionaCapitoloComeLetto(idCapitolo:  String)  +aggiungiACollezione(idLibro: String)  +rimuoviDaCollezione(idLibro: String)  +aggiungiTrial(idLibro: String) | |

La scelta di utilizzare StatoUtente e TipoUtente è stata influenzata dal *Pattern State*.



Scrittore

-documento: File

+getDocumento(): File

+caricaStruttura(): void

+getLibri(): List<Libro>

L'esigenza era quella di cambiare classe a tempo di esecuzione, cosa ovviamente impossibile con la specializzazione mediante estensione che agisce a tempo di compilazione.

*Diagramma di Dettaglio: Dominio – Libri*



\* -\_transazioni

1 -\_utente

\* -\_libri

*Transazione*

1 -\_scrittore

-titolo : String

+getTitolo(): String

+getCapitoli(): List<Capitolo>

+getScrittore(): Scrittore

+getTipoLibro(): TipoLibro

+setTipoLibro(TipoLibro): void

+getStatoLibro(): StatoLibro

+getId(): String

+aggiungiTestoIntegrale(testoIntegrale: File): void

+sponsorizza(): void

Capitolo

#tipo: String

#timestamp: LocalDateTime

+getTipo(): String

+getTimeStamp: LocalDateTime

+getUtente(): Utente

1..n

1

\*

-\_capitoli

-\_libro

-titolo: String

+getTitolo(): String

+getCommenti(): List<Commento>

+getValutazioni(): List<Valutazione>

+getLibro(): Libro

+getId(): String

+valutaCapitolo(valutazione: Valutazione): void

+assegnaEmojiCapitolo(emoji: Emoji): void

+commentaCapitolo(commento: String): void

complete,disjoint

-\_libro

1 Sponsorizzazione

#tipo: String

#timestamp: LocalDateTime

+getLibro(): Libro

PassaggioAPremium

-\_statoLibro

#tipo: String

#timestamp: LocalDateTime

-\_tipoLibro 1

1

-\_libriInLettura \*

-\_libro 1

1 -\_libro

1

*StatoLibro*

#statoLibro: String

#TimeStampUpdate: LocalDateTime

+getStatoLibro(): String

+getTimeStampUpdate: LocalDateTime

+getLibro(): Libro

0..3 -\_libriInTrial

LibroNonSponsorizzato

LibroSponsorizzato

\*

#statoLibro: String

#TimeStampUpdate: LocalDateTime

#statoLibro: String

#TimeStampUpdate: LocalDateTime

-\_sponsorizzazione 1

#tipoLibro: String

#TimeStampUpdate: LocalDateTime

#tipoLibro: String

#TimeStampUpdate: LocalDateTime

-testoIntegrale: File

+leggi(): File

LibroNonInTrial

LibroInTrial

#tipoLibro: String

#TimeStampUpdate: LocalDateTime

+getTipoLibro(): String

+getTimeStampUpdate: LocalDateTime

+getLibro(): Libro

*TipoLibro*

+getLibriInLettura(): List<Libro>

+getLibroInTrial(): LibroInTrial

+selezionaCapitoloComeLetto(idCapitolo: String)

+aggiungiACollezione(idLibro: String)

+rimuoviDaCollezione(idLibro: String)

+aggiungiTrial(idLibro: String)

Lettore

Libro

-documento: File

+getDocumento(): File

+caricaStruttura(): void

+getLibri(): List<Libro>

Scrittore

-nome: String

-cognome: String

+getNome(): String

+setNome(): void

+getCognome(): String

+setCognome(): void

+getStatoUtente(): StatoUtente

+setStatoUtente(StatoUtente): void

+getTipoUtente(): TipoUtente

+setTipoUtente(TipoUtente): void

+getTransazioni(): List<Transazione>

*Utente*

Come anche nel caso di StatoUtente e TipoUtente, gli stessi principi sono stati utilizzati per rappresentare libri in trial e libri sponsorizzati.

*Diagramma di Dettaglio: Dominio – Log*

|  |  |
| --- | --- |
| <<Enumeration>>  Livello | |
| +Basso  +Medio  +Alto | |
| \*  1 |  |
| Voce di Log | |
| -timestamp: DateTime  -descrizione: String | |
|  | |
|  |  |
| Log | |
|  | |
| +visualizzaLog()  +aggiungiLog(VoceDiLog) | |

*Diagramma di Dettaglio: Interfacce*

|  |
| --- |
| <<Interface>>  ILogin |
| +verificaCredenziali(Username: String,  Password: String) |

|  |
| --- |
| <<Interface>>  IGestioneAmministratore |
|  |
| +getAssistenti(): List<Assistente>  +creazioneAccountAssistenza(Username:  String, Password: String)  +rimozioneAccountAssistenza(Username:  String) |

|  |
| --- |
| <<Interface>>  IGestioneLettore |
|  |
| +getUsername(): Username: String |

|  |
| --- |
| <<Interface>>  ISchedaLibro |
|  |
| +getCommenti(IdCapitolo: String)  +aggiungiACollezione(Username: String,  IdLibro: String)  +rimuoviDaCollezione(Username: String, IdLibro: String)  +aggiungiTrial(Username: String, IdLibro: String)  +selezionaCapitoloComeLetto(Username: String, IdCapitolo: String)  +commentaCapitolo(Username: String,  IdCapitolo: String, Commento: String)  +valutaCapitolo(Username: String,  IdCapitolo: String, Valutazione: Integer)  +reagisciACapitolo(Username: String,  IdCapitolo: String, Emoji: String) |

|  |
| --- |
| <<Interface>>  IRegistrazione |
|  |
| +registraUtente(Nome: String, Cognome: String, Username: String, Password: String)  +registraUtente(Nome: String, Cognome: String, Username: String, Password: String, DocumentoDiIdentità: File) |

|  |
| --- |
| <<Interface>>  IGestioneAssistenza |
|  |
| +getScrittoriDisabilitati(): List<Scrittore>  +getUtentiBloccati(): List<Utente>  +getSegnalazioni(): List<Segnalazione>  +bloccaUtente(Username: String)  +sbloccaUtente(Username: String)  +rimuoviCommento(IdCommento: String)  +accettaScrittore(Username: String) |

|  |
| --- |
| <<Interface>>  ILog |
|  |
| +getVociDiLog(DateTime, DateTime):  List<VoceDiLog>  +getVociDiLog(Date): List<VoceDiLog> |

|  |
| --- |
| <<Interface>>  IGestioneScrittore |
|  |
| +getUsername(): Username: String |

|  |
| --- |
| <<Interface>>  IPassaggioAPremium |
|  |
| +passaAPremium(Username: String) |

|  |
| --- |
| <<Interface>>  ILogger |
|  |
| +aggiungiLog(VoceDiLog) |

|  |
| --- |
| <<Interface>>  IInserimentoLibro |
|  |
| +inserisciLibro(Username: String, Titolo:  String, Capitoli: List<Capitolo>)  +inserisciLibro(Username: String, Titolo: String, Capitoli: List<Capitolo>, Testo: File)  +aggiungiTestoIntegrale(Username:  String, IdLibro: String, Testo: File) |

|  |
| --- |
| <<Interface>>  IRicercaLibro |
|  |
| +getLibri(Titolo: String, Tipo: String):  List<String> |

|  |
| --- |
| <<Interface>>  ILettura |
|  |
| +visualizzaTesto(Username: String, IdLibro: String): File |

|  |
| --- |
| <<Interface>>  ISponsorizzazione |
|  |
| +sponsorizza(Username: String, IdLibro: String) |

|  |
| --- |
| <<Interface>>  ISegnalazione |
|  |
| +segnalaCommento(Username: String,  IdCommento: String, Motivazione: String) |

*Diagramma di Dettaglio: Lettore*

IRicercaLibro

IGestioneLettore

IPassaggioAPremium

RicercaLibro Controller

GestioneLettore Controller

1

1

PassaggioAPremium Controller

1 1

1

1

1

ISegnalazione

ISchedaLibro

1

ILettura

1

SchedaLibro Controller

1

+visualizzaTesto(Username: String, IdLibro: String): File

+getCommenti(IdCapitolo: String)

+aggiungiACollezione(Username: String, IdLibro: String)

+rimuoviDaCollezione(Username: String,

IdLibro: String)

+aggiungiTrial(Username: String, IdLibro: String)

+selezionaCapitoloComeLetto(Username:

String, IdCapitolo: String)

+commentaCapitolo(Username: String, IdCapitolo: String, Commento: String)

+valutaCapitolo(Username: String,

IdCapitolo: String, Valutazione: Integer)

+reagisciACapitolo(Username: String, IdCapitolo: String, Emoji: String)

+segnalaCommento(Username: String, IdCommento: String, Motivazione: String)

Lettura Controller

Segnalazione Controller

+passaAPremium(Username: String)

+getUsername(): Username: String

+getLibri(Titolo: String, Tipo: String):

List<String>

-openConnection(String) #Connection getConnection()

Controller

-dbConnection: Connection

*Diagramma di Dettaglio: Scrittore*



IInserimentoLibro

InserimentoLibro Controller

IGestioneScrittore

ISponsorizzazione

1

1

1

1

+getUsername(): Username: String

+sponsorizza(Username: String, IdLibro: String)

+inserisciLibro(Username: String, Titolo: String, Capitoli: List<Capitolo>)

+inserisciLibro(Username: String, Titolo: String, Capitoli: List<Capitolo>, Testo: File)

+aggiungiTestoIntegrale(Username: String, IdLibro: String, Testo: File)

Sponsorizzazione Controller

GestioneScrittore Controller

-openConnection(String) #Connection getConnection()

Controller

-dbConnection: Connection

*Diagramma di Dettaglio: Amministratore*

IGestioneAmministratore

GestioneAmministratore Controller

+getAssistenti(): List<Assistente>

+creazioneAccountAssistenza(Username: String, Password: String)

+rimozioneAccountAssistenza(Username: String)

1

ILog

1

Log Controller

logReader: Reader

+getVociDiLog(DateTime, DateTime)

+getVociDiLog(Date)

-openConnection(String)

#Connection getConnection()

Controller

-dbConnection: Connection

*Diagramma di Dettaglio: Assistenza*

IGestioneAssistenza

GestioneAssistenza Controller

+getScrittoriDisabilitati(): List<Scrittore>

+getUtentiBloccati(): List<Utente>

+getSegnalazioni(): List<Segnalazione>

+bloccaUtente(Username: String)

+sbloccaUtente(Username: String)

+rimuoviCommento(IdCommento: String)

+accettaScrittore(Username: String)

-openConnection(String) #Connection getConnection()

Controller

-dbConnection: Connection

Si è fattorizzata nella classe Controller tutta la gestione della persistenza dei dati in

modo da non dover replicare la scrittura e lettura su DB. Solo i controller di gestione ereditano da Controller, gli altri controller accedono ai servizi di Controller attraverso le associazioni bi-direzionali.

+aggiungiLog(VoceDiLog)

Logger

logWriter: Writer

Ogni controller utilizza il Logger per la scrittura dei Log, questa relazione con tutti i controller non è stata riportata nei diagrammi precedenti in quanto avrebbe reso gli stessi più pesanti.

*Diagramma di Dettaglio: Broker*

1

1

+gestisciRichiesta(richiesta: Richiesta): Richiesta

FiltroRichieste

SessioneAttiva

-chiave: String

* ruolo: String
* dataCreazione: Date
* dataScadenza: Date
* ipClient: String

Sessione

|  |
| --- |
| <<singleton>>  GestoreSessione |
|  |
| +creaSessione(sessione: Sessione): String  +verificaSessione(idSessione:String): Sessione |

|  |  |
| --- | --- |
|  | |
| Richiesta |  |
| * destinatario: String * mittente: String * servizio: String * parametri: Array(Object) |
|  |
|  |

Qui si ha la struttura del Broker:

l’entry point è FiltroRichieste, che manipola le richieste che

arrivano dai client aggiungendo la sessione e verificando che la sessione sia attiva, smista poi le richieste al server corretto e restituisce le risposte ai Client.

*Diagramma di Dettaglio: View rese disponibili ai Client*

+segnalaCommento(String, String, String)

View Segnalazione

+ List<String> getLibri(String, String)

+aggiungiACollezione(String, String)

+rimuoviDaCollezione(String, String)

+aggiungiTrial(String, String)

+selezionaCapitoloComeLetto(String, String)

+commentaCapitolo(String, String, String)

+valutaCapitolo(String, String, Integer)

+reagisciACapitolo(String, String, String)

View RicercaLibri

View SchedaLibro

+passaAPremium(String)

+ String getUsername()

View PassaggioAPremium

Home GestioneLettore

+sponsorizza(String, String)

+inserisciLibro(String, String, List<String>)

+inserisciLibro(String, String, List<String>, File)

+aggiungiTestoIntegrale(String, String, File)

View Sponsorizzazione

View InserimentoLibro

+ String getUsername()

Home GestioneScrittore

+ File visualizzaTesto(String, String)

View Lettura

+rimuoviCommento(String)

+ List<String> getScrittoriDisabilitati()

+accettaScrittore(String)

+bloccaUtente(String)

+sbloccaUtente(String)

View AccettazioneScrittore

View RimozioneCommento

View GestioneBan

+ List<String> getUtentiBloccati()

+getSegnalazioni()

Home GestioneAssistenza

+creazioneAccountAssistenza(String, String)

+rimozioneAccountAssistenza(String)

+ List<VoceDiLog> getVociDiLog(DateTime, DateTime)

+ List<VoceDiLog> getVociDiLog(Date)

View Controllo Assistenza

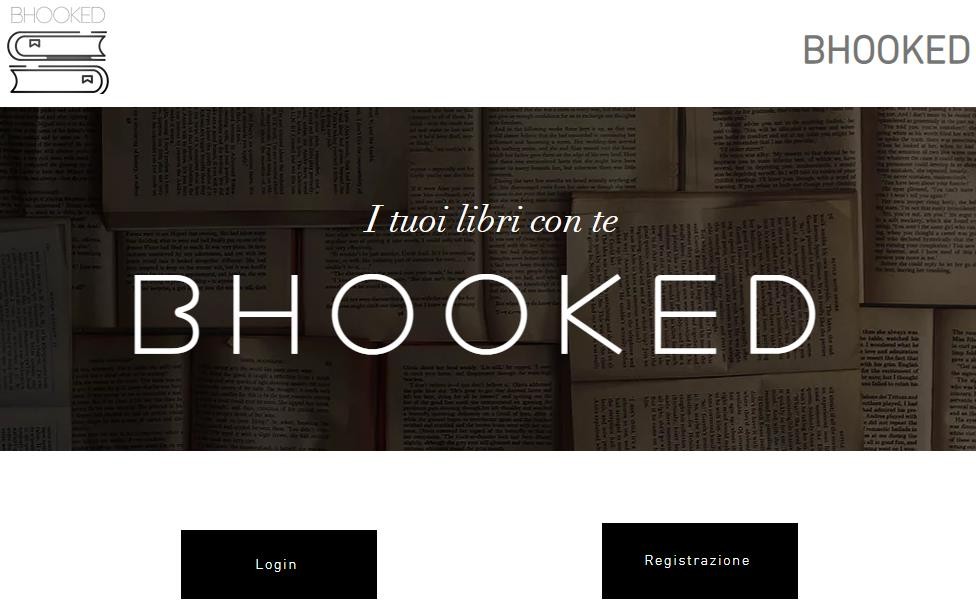
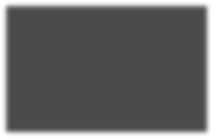
View Visualizzazione Log

+ String getAssistenti()

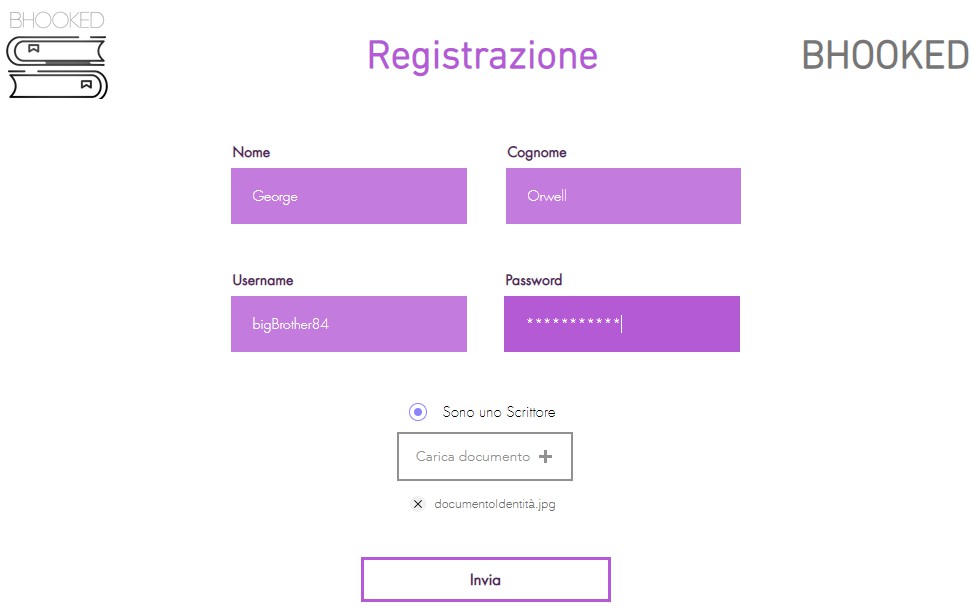
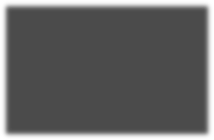
Home Gestione Amministratore

Alcune interfacce

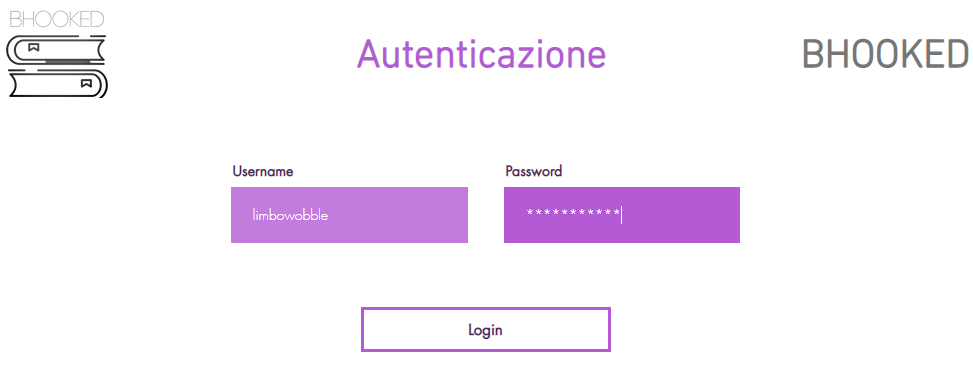
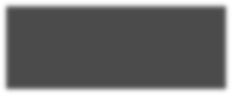
*Welcome page*



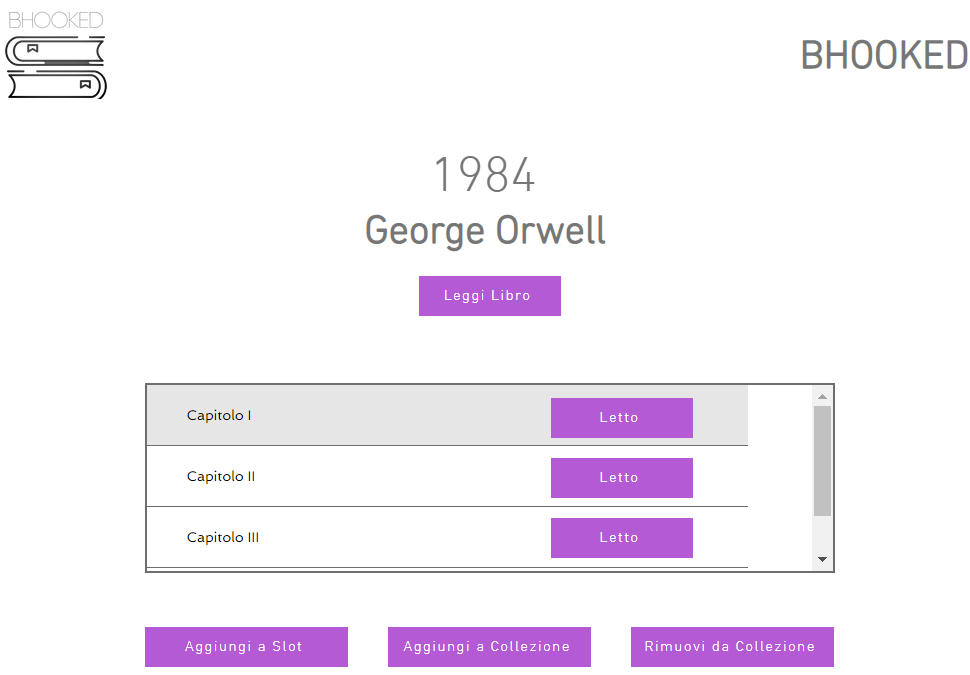
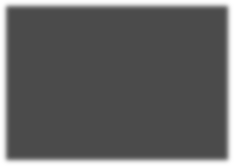
*View Registrazione*



*View Login*

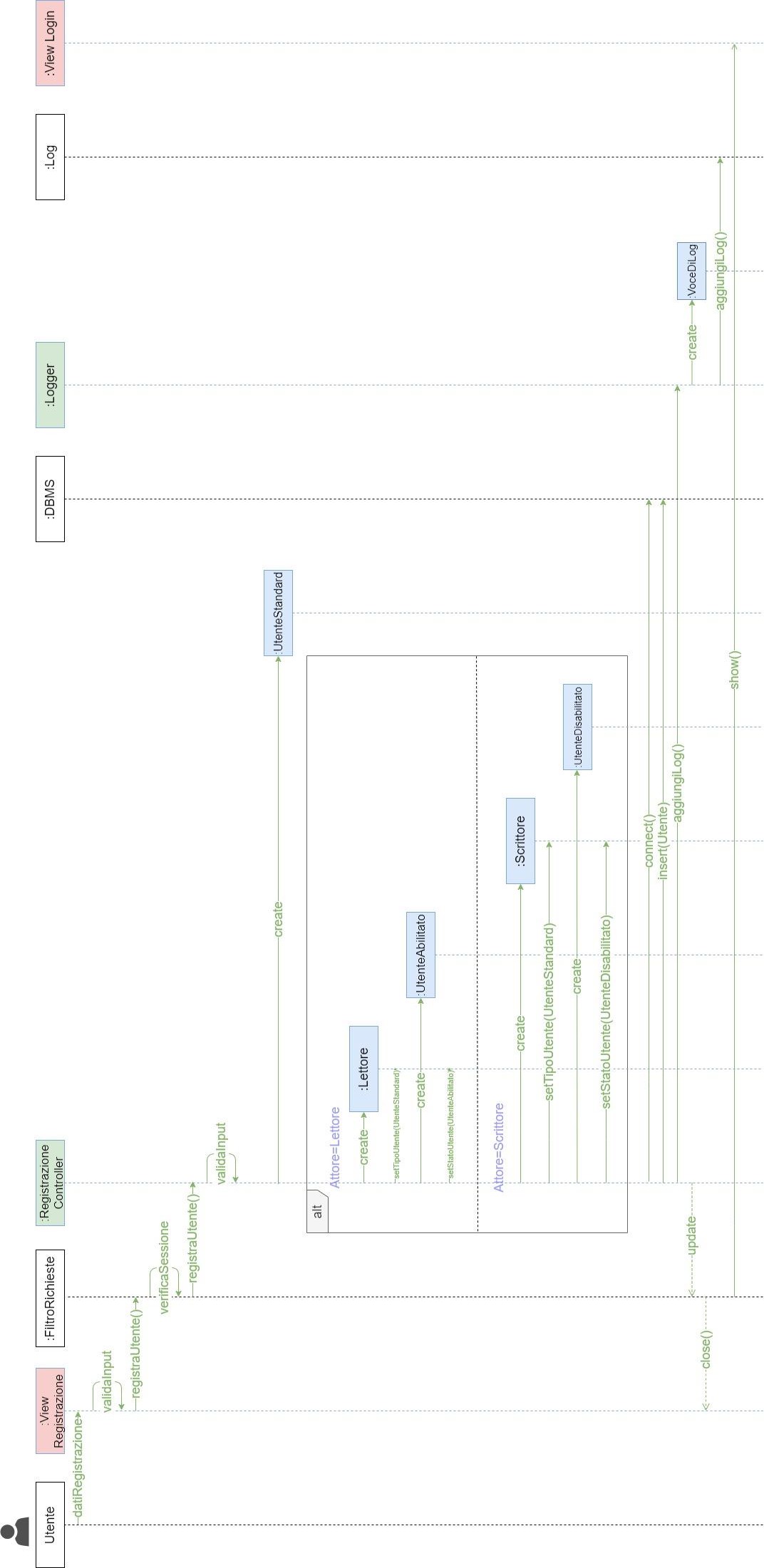


*View Scheda del Libro*



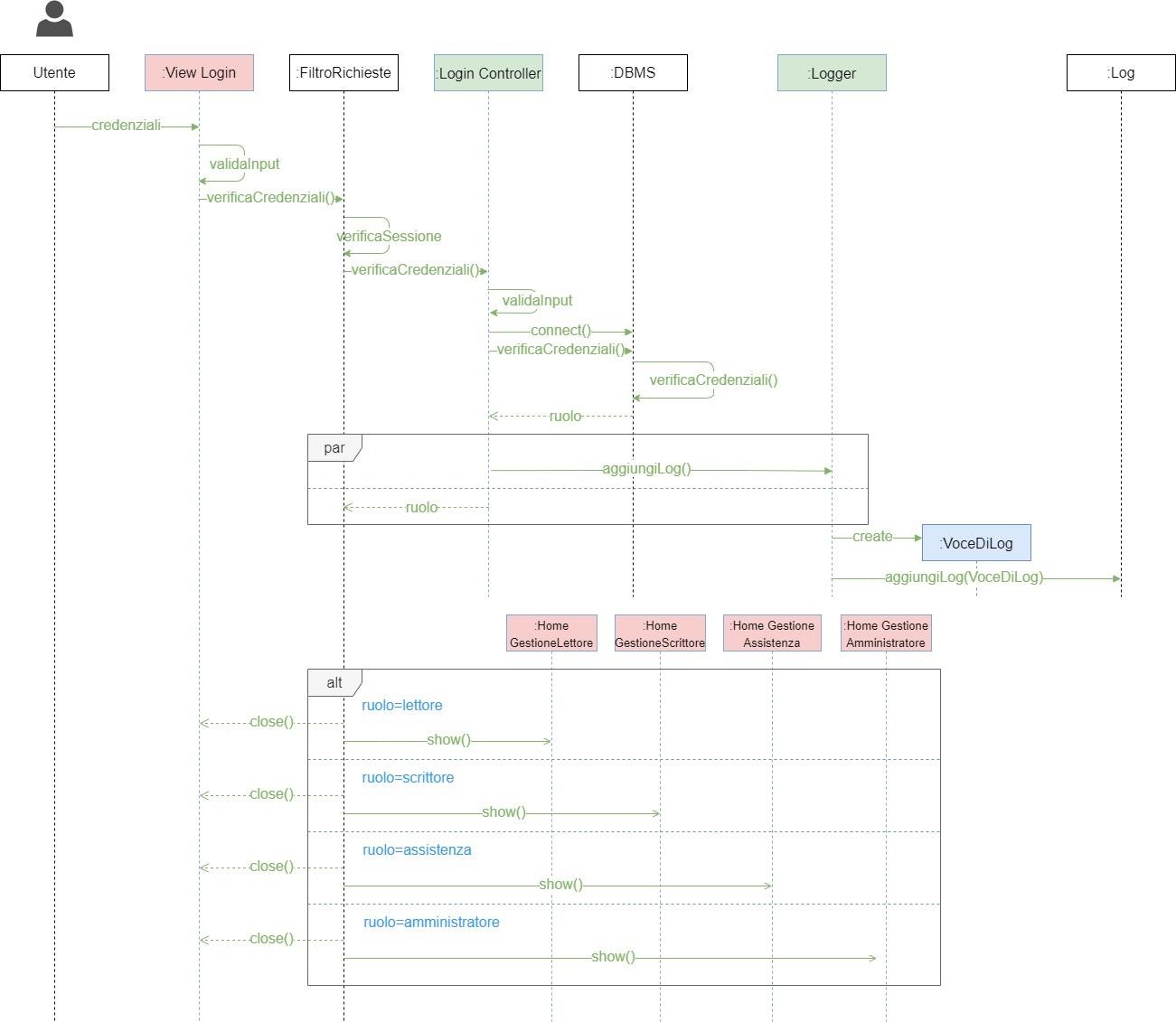
##### Interazione

*Diagramma di sequenza: Registrazione*



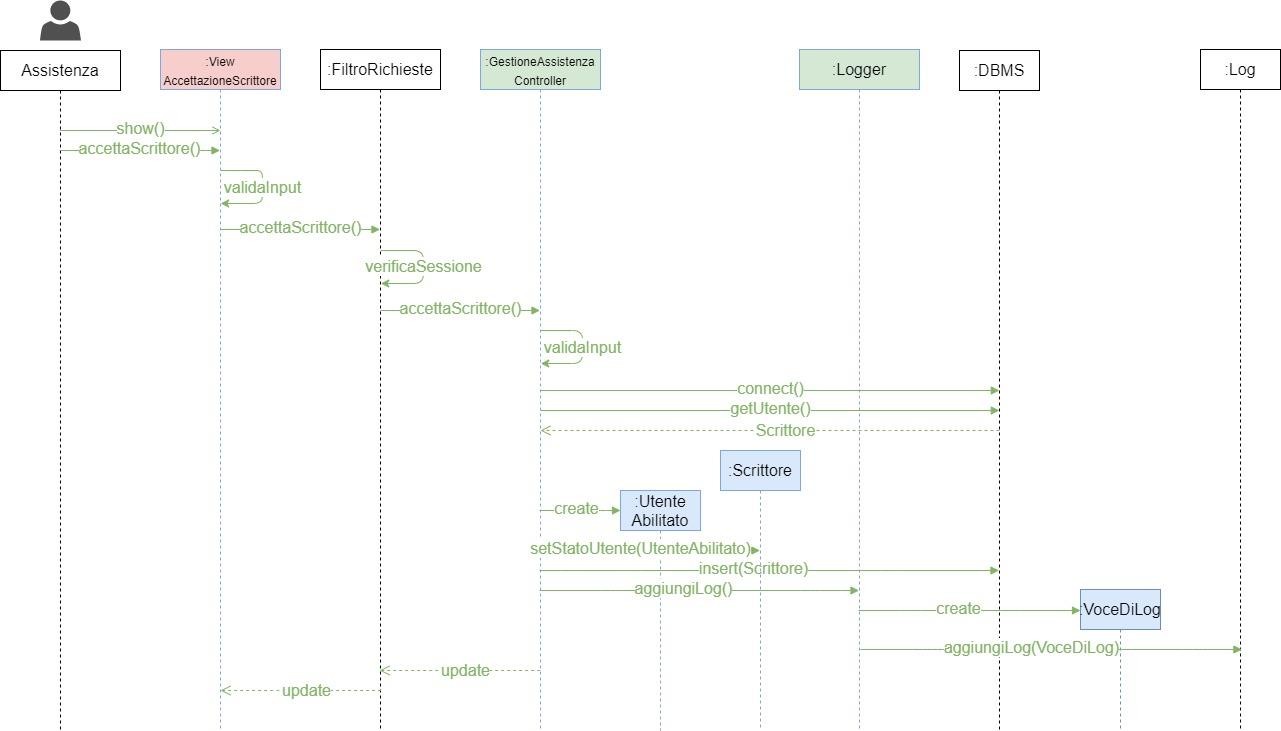
Nella fase di registrazione viene distinto il caso del lettore da quello dello scrittore. Infatti, la differenza sta sia nella creazione dell’istanza e del tipo dell’utente.

*Diagramma di sequenza: Login*



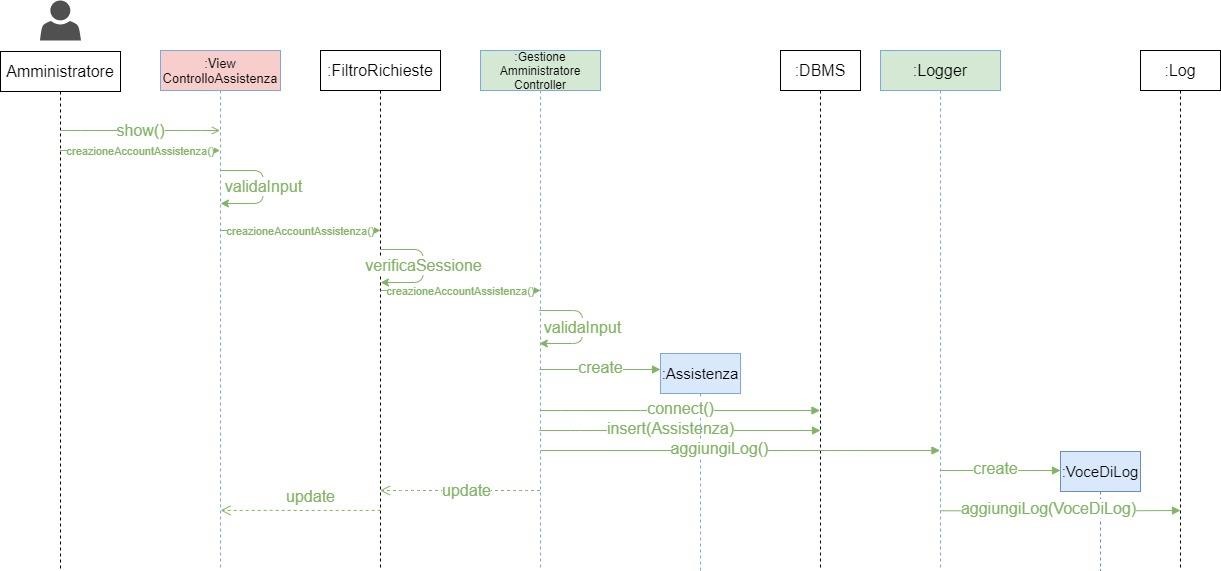
Un aspetto da mettere in evidenza è che facendo l’accesso il Database restituisce il ruolo associato ad un certo account in modo da mostrare la Home adatta al ruolo ricoperto da chi si sta autenticando.

*Diagramma di sequenza: Accettazione Scrittore*



L’istanza di scrittore è già presente nel Database al momento dell’accettazione da parte dell’Assistenza o dell’Amministratore. Per questo viene ottenuta l’istanza di Scrittore e si crea un nuovo stato dell’Utente (abilitato) che verrà sostituito al precedente stato (disabilitato) che è quello con cui di default viene creato uno Scrittore. Al termine di questa operazione si aggiunge il log in modo persistente e si aggiorna la view dell’assistente da cui viene rimosso lo scrittore appena accettato.

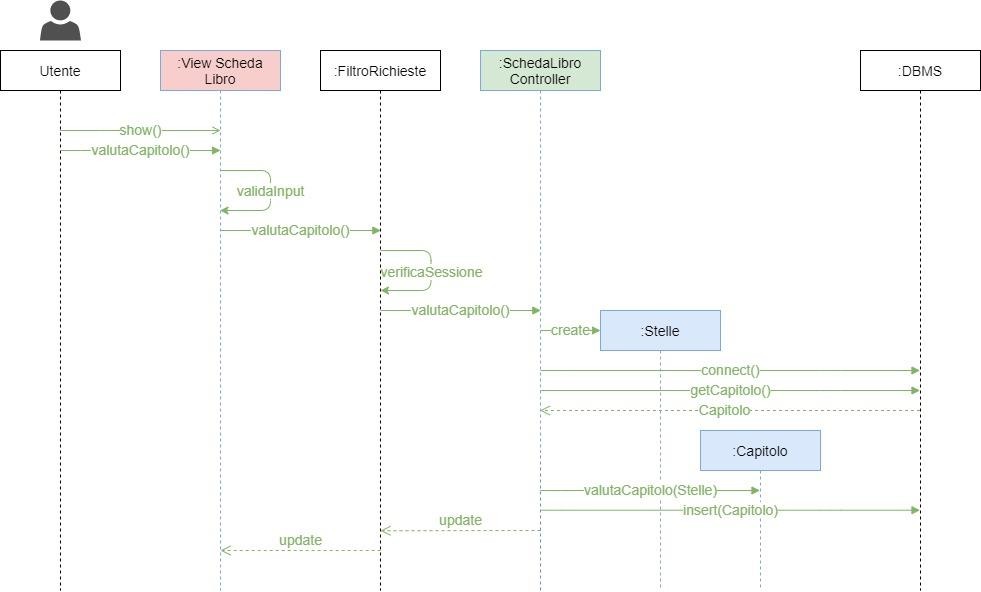
*Diagramma di sequenza: Creazione Account Assistenza*



La creazione di un account assistenza è molto semplice, quindi le interazioni tra i componenti

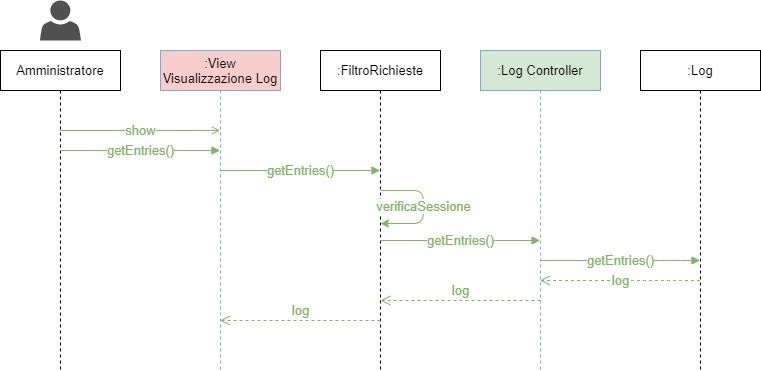
dell’applicazione sono minime.

*Diagramma di sequenza: Valutazione Capitolo*



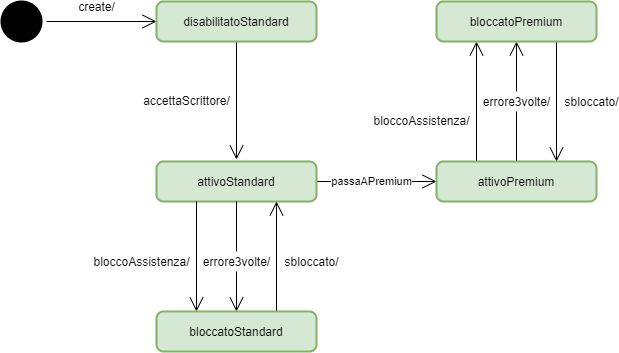
Per la valutazione si crea un’istanza della classe Stelle che viene usata come parametro per “valutaCapitolo”. In futuro, nel caso in cui ci siano altre forme di valutazione, rimarrà tutto uguale tranne la creazione dell’istanza di Stelle, si creerà, infatti, un’istanza del tipo specificato sulla base della nuova forma di valutazione stessa.

*Diagramma di sequenza: Visualizzazione Log*



##### Comportamento

*Diagramma di Stato: Scrittore*



Rispetto all’analisi è stato aggiunto lo stato UtenteDisabilitato.

## Progettazione della persistenza

Nel database vengono salvate le SaltedHashPassword che permettono di evitare attacchi di tipo Rainbow Tables mantenendo in memoria non il semplice hash della password ma l’hash della concatenazione di un numero pseudorandomico con la password dell’utente. Per fare ciò si utilizza l’algoritmo SHA256

La relazione slot tra LETTORI e LIBRI\_IN\_TRIAL non riesce a rappresentare un vincolo: nel caso in cui il valore di premiumTimestamp (nell’utente) sia nullo, allora la cardinalità 0-3 deve essere ridotta a 0-1 in quanto un utente standard può occupare al massimo uno slot con un libro che presenta il testo integrale.

## Progettazione del collaudo

public class **TestRicercaLibro**{ private DataBaseMock db;

@BeforeAll

public void **setup**(){

db = new **DataBaseMock**();

*// Dati scrittore*

String username = "Orwell"; String nome = "George"; String cognome = "Orwell";

*// Inserimento dati in scrittore*

Scrittore scrittore = new **Scrittore**(username, nome, cognome); StatoUtente statoScrittore = new **UtenteAbilitato**(scrittore); TipoUtente tipoScrittore = new **UtenteStandard**(scrittore); scrittore.**setStatoUtente**(statoScrittore); scrittore.**setTipoUtente**(tipoScrittore);

*// Creazione ed inserimento dati immodificabili in libro*

String titolo = "1984";

List<String> capitoli = new ArrayList<>(); indice.**add**("Capitolo1"); indice.**add**("Capitolo2");

Libro libro = new **Libro**(scrittore, titolo, capitoli);

*// Creazione dati dinamici del libro*

StatoLibro statoLibro = new **LibroNonSponsorizzato**(libro); TipoLibro tipoLibro = new **LibroNonInTrial**(libro);

*// Inserimento dati dinamici del libro* libro.**setStatoLibro**(statoLibro); libro.**setTipoLibro**(tipoLibro);

*// Aggiunta libro allo scrittore*

scrittore.**aggiungiLibro**(libro);

*/\*simulazione dell'interazione con db che*

*nell'implementazione reale avviene per mezzo di query sql \*/*

db.**addScrittore**(scrittore);

*// Dati scrittore*

username = "EgidioAlanoPoe"; nome = "Edgar Allan"; cognome = "Poe";

*// Inserimento dati in scrittore*

scrittore = new **Scrittore**(username,nome,cognome); statoScrittore = new **UtenteDisabilitato**(scrittore); tipoScrittore = new **UtenteStandard**(scrittore); scrittore.**setStatoUtente**(statoScrittore); scrittore.**setTipoUtente**(tipoScrittore);

*/\*simulazione dell'interazione con db che*

*nell'implementazione reale avviene per mezzo di query sql \*/*

db.**addScrittore**(scrittore);

*// Dati lettore* username = "MarioRed96"; nome = "Mario";

cognome = "Rossi";

*// Inserimento dati in lettore*

Lettore lettore = new **Lettore**(username, nome, cognome);

*/\*simulazione dell'interazione con db che*

*nell'implementazione reale avviene per mezzo di query sql \*/*

db.**addLettore**(lettore);

*// Dati lettore* username = "BookLover83"; nome = "Paolo";

cognome = "Bianchi";

*// Inserimento dati in lettore*

lettore = new **Lettore**(username, nome, cognome);

*/\*simulazione dell'interazione con db che*

*nell'implementazione reale avviene per mezzo di query sql \*/*

db.**addLettore**(lettore);

}

@Test

public void **testRicerca**(){ String titolo = "1984"; String tipo = "Tutti";

RicercaLibroController controllerRicercaLibro = new **RicercaLibroController**();

List<Libro> libri = controllerRicercaLibro.**getLibri**(titolo,tipo);

**assertEquals**(titolo, libri.**get**(0).**getTitolo**());

}

@Test

public void **testEmoji**(){

String username = "RagazzoPrudente";

String idCapitolo = "Orwell@1984@Capitolo24";

String nomeEmoji = "Confuso";

SchedaLibroController schedaLibroController = new **SchedaLibroController**();

schedaLibroController.**reagisciACapitolo**( username,idCapitolo,nomeEmoji);

*/\*simulazione dell'interazione con db che*

*nell'implementazione reale avviene per mezzo di query sql \*/*

Map<Emoji,List<Utente>> emojiUtenti = db.**getEmojiCapitolo**(idCapitolo);

List<Utente> utenti = emojiUtenti.**get**(nomeEmoji);

Boolean isPresent = false; for(Utente u: utenti){

if(u.getUsername.**compareTo**(username)==0) { isPresente = true;

break;

}

}

**assertEquals**(true,isPresente);

}

@Test

public void **testCommento**(){

String username = "MarioRed96";

String idCapitolo = "Orwell@1984@Capitolo1";

String commento = "Questo capitolo è stato particolarmente avvincente. " + "Alla fine non va a finire bene. " +

"Nell'ultimo capitolo il protagonista cambia idea";

SchedaLibroController schedaLibroController = new **SchedaLibroController**();

schedaLibroController.**commentaCapitolo**( username, idCapitolo, commento);

List<Commento> commenti = schedaLibroController.**getCommenti**(idCapitolo);

**assertEquals**(commento, commenti.**get**(commenti.**size**()-1));

}

@Test

public void **testCreazioneAccountAssistenza**(){ String username = "RagazzoPrudente"; String password = "SilenzioBruno";

GestioneAmministratoreController gestioneAmministratoreController = new **GestioneAmministratoreController**();

gestioneAmministratoreController.**creaAccountAssistenza**(username,password); List<Assistenza> assistenti =

gestioneAmministratoreController.**getAssistenti**();

**assertEquals**(username,assisenti.**get**(assistenti.**size**()-1).**getUsername**());

}

@Test

public void **testAccettazioneScrittore**(){ String username = "EgidioAlanoPoe";

GestioneAssistenzaController gestioneAssistenzaController = new **GestioneAssistenzaController**();

List<Scrittore> scrittoriDisabilitatiPrima = gestioneAssistenzaController.**getScrittoriDisabilitati**();

gestioneAssistenzaController.**accettaScrittore**(username); List<Scrittore> scrittoriDisabilitatiDopo =

gestioneAssistenzaController.**getScrittoriDisabilitati**();

**assertNotEquals**(scrittoriDisabilitatiDopo,scrittoriDisabilitatiPrima);

}

@Test

public void **testSegnalazione**(){

String usernameUtenteSegnalante = "RagazzoPrudente"; String idCommento = "Orwell@1984@Capitolo1@MarioRed96@" +

LocalDateTime.**now**().**toString**(); String motivazione = "Spoiler del finale";

SegnalazioneController segnalazioneController = new **SegnalazioneController**();

segnalazioneController.**segnalaCommento**( usernameUtenteSegnalante,idCommento, motivazione);

GestioneAssistenzaController gestioneAssistenzaController = new **GestioneAssistenzaController**();

List<Segnalazione> segnalazioni = gestioneAssistenzaController.**getSegnalazioni**();

**assertEquals**(

usernameUtenteSegnalante, segnalazioni.**get**(

segnalazioni.**size**()-1).**getUtenteSegnalante**().**getUsername**());

**assertEquals**(

motivazione,

segnalazioni.**get**(segnalazioni.**size**()-1).**getMotivazione**());

**assertEquals**(

idCommento,

segnalazioni.**get**(segnalazioni.**size**()-1).**getCommento**().**getId**());

}

@Test

public void **testBlocco**(){

String username = "MarioRed96";

GestioneAssistenzaController gestioneAssistenzaController =

new **GestioneAssistenzaController**(); gestioneAssistenzaController.**bloccaUtente**(username);

*/\*simulazione dell'interazione con db che*

*nell'implementazione reale avviene per mezzo di query sql \*/* Utente utente = db.**getUtente**(username); **assertEquals**("UtenteBloccato",utente.**getStato**().**getNomeStato**());

}

}

### Deployment

|  |  |
| --- | --- |
| ServerLettore | |
| <<man | ifest>> |
| ServerLettore.war | |

|  |  |
| --- | --- |
| ServerScrittore | |
| <<man | ifest>> |
| ServerScrittore.war | |



<<manifest>>

ServerAssistenza.war

ServerAssisstenza



<<manifest>>

ServerAmministratore.war

ServerAmministratore



Broker

<<manifest>>

Broker.war

|  |  |
| --- | --- |
| Log | |
| <<man | ifest>> |
| Log.war | |

\*

1

ServerRegistrazione

<<deploy>> <<deploy>>

Login.war

Broker.war

Client



<<deploy>>

Log.war

<<deploy>>

<<deploy>>

<<deploy>>

<<deploy>>

ServerAssistenza.war

1

1

1

ServerRegistrazione

ServerLettore.war

ServerAmministratore.war

ServerScrittore.war

DBMS

1

1

1

1

1

1

1

1

1

LogFisico

1

ServerLettore

1

ServerScrittore

1

ServerAssistenza

ServerAmministratore