
Software Requirements Specification

for

BookExchange

Version 1.0 approved

Prepared by Pasquale Cioffi

Università degli Studi della Campania Luigi Vanvitelli

Table of Contents

Table of Contents	ii
1. Introduction	1
1.1 Purpose	1
1.2 Product Scope	1
1.3 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	3
2.3 User Classes and Characteristics	4
2.4 Operating Environment	5
2.5 Design and Implementation Constraints	5
2.6 Diagrammi di progettazione.....	2
3. External Interface Requirements	7
3.1 User Interfaces	8
3.2 Software Interfaces.....	10
3.3 Communications Interfaces	11
4. System Features	12
4.1 Ricerca di libri tramite GoogleBooks.....	Errore. Il segnalibro non è definito.
4.2 Ricerca di libri posseduti dagli utenti del sistema	Errore. Il segnalibro non è definito.
4.3 Scambio di libri tra utenti	Errore. Il segnalibro non è definito.
4.4 Scambio di messaggi tra utenti.....	Errore. Il segnalibro non è definito.
4.5 Registrazione utente	Errore. Il segnalibro non è definito.
5. Other Nonfunctional Requirements	5
5.1 Performance Requirements.....	15
5.3 Security Requirements.....	15
5.4 Software Quality Attributes	15

1. Introduction

1.1 Purpose

Creare un'applicazione web che permetta agli utenti di scambiarsi libri usati. L'applicazione dovrebbe consentire agli utenti di elencare i libri che desiderano scambiare e cercare libri disponibili per lo scambio. L'applicazione deve anche gestire i dettagli degli utenti, i feedback sugli scambi e fornire un sistema di messaggistica per la comunicazione tra gli utenti.

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2 Product Scope

L'applicazione deve permettere di consultare il catalogo di libri disponibili messi a disposizione dagli altri utenti iscritti tramite ricerca per autore, titolo, categoria, etc.

Gli utenti iscritti alla piattaforma potranno inserire le informazioni dei libri che posseggono e che intendono scambiare, specificandone il titolo, l'autore, l'immagine di copertina, etc.

Tramite la funzione di ricerca, l'utente potrà cercare il libro che gli interessa, contattare il proprietario del libro e proporre uno scambio con uno dei suoi libri. Una volta confermato lo scambio, i due utenti potranno rimanere in contatto tramite un servizio di messaggistica istantanea con la quale potranno concordarsi sullo scambio materiale del libro. Avvenuto lo scambio gli utenti potranno confermare la transazione tramite l'applicazione web e i loro cataloghi verranno aggiornati di conseguenza. In un qualsiasi momento l'offerente potrà rifiutare la richiesta di scambio e l'utente che l'ha proposto potrà annullarla.

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.3 References

Per lo styling di tutte le pagine web dell'applicazione è stato utilizzato [Bootstrap](https://getbootstrap.com/) (in particolare la versione 5.31), un framework gratuito di CSS.

<https://getbootstrap.com/>

Inoltre per agevolare l'inserimento dei dati dei libri da parte degli utenti è stata utilizzata l'API [GoogleBooks](https://developers.google.com/books/docs/overview?hl=it) (v1) offerta da Google gratuitamente per le query di ricerca.

<https://developers.google.com/books/docs/overview?hl=it>

2. Overall Description

2.1 Product Perspective

Per la realizzazione di quest'applicazione web sono stati individuati i seguenti requisiti:

Requisiti funzionali:

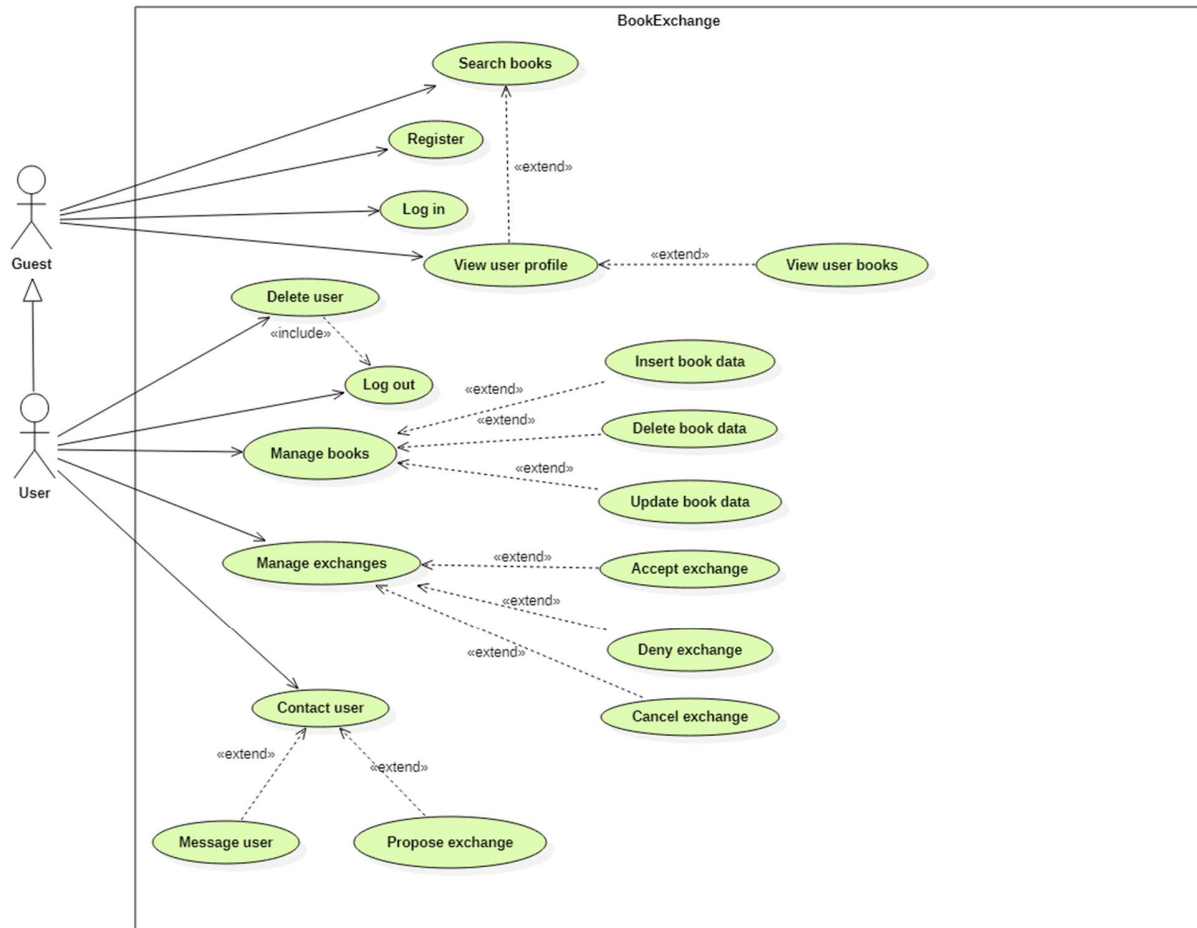
- Il sistema deve permettere all'utente di registrarsi tramite la scelta di uno username, email e password validi.
- Il sistema deve permettere all'utente registrato di accedervi con le sue credenziali (email, password), mantenendo la sua sessione attiva.
- Il sistema deve permettere all'utente di inserire le informazioni dei libri che possiede e intende scambiare.
- Il sistema deve permettere all'utente di visualizzare/modificare/eliminare i dati sui libri che ha inserito.
- Il sistema deve permettere all'utente, registrato o meno (ospite), di ricercare tra i libri disponibili il libro interessato.
- Il sistema deve permettere all'utente di proporre uno scambio con un altro utente che ha inserito nel suo catalogo il libro interessato.
- Il sistema deve permettere all'utente di comunicare tramite sistema di messaggistica con gli altri utenti del sistema.

Requisiti non funzionali:

- Il sistema deve essere facile da utilizzare e intuitivo per l'utente finale
- Il sistema deve facilitare l'inserimento delle informazioni dei libri tramite auto-completamento.
- Il sistema deve garantire la sicurezza sui dati forniti dai suoi utenti.

2.2 Product Functions

Di seguito è riportato il diagramma dei casi d'uso che descrive le funzionalità principali dell'applicazione dal punto di vista dell'utente:



Invece di seguito è riportato il diagramma dei casi d'uso dal punto di vista di un client dell'API fornita dall'applicazione in formato REST



2.3 User Classes and Characteristics

Gli attori identificati che andranno ad utilizzare l'applicazione sono i seguenti:

- Guest (ospite): rappresenta un utente che non ha registrato un account nel sistema o che non ha ancora effettuato l'accesso.
- User (Utente): rappresenta un utente registrato che ha effettuato l'accesso al sistema.
- REST Client : rappresenta un qualsiasi client che effettua richieste HTTP all'API del sistema

2.4 Operating Environment

Il back-end del sistema è stato sviluppato tramite **PHP**, e si occupa di gestire le richieste all'API e di fornire le pagine web richieste dai client.

Il front-end del sistema è stato sviluppato interamente in **HTML** e **JavaScript** e stilizzato in **CSS** tramite il framework **Bootstrap**.

Il codice HTML rappresenta la struttura della pagina web richiesta e viene fornito dal server tramite PHP, mentre in JavaScript è presente la logica per l'interazione con la pagina web, per la validazione dei dati e per inviare le eventuali richieste **HTTP**.

Il sistema opera tramite un **Database SQL** del tipo **MariaDB**, il server è hostato in locale tramite l'applicazione **XAMPP** e **phpMyAdmin** che si occupa della virtualizzazione del server e di gestire le richieste.

2.5 Design and Implementation Constraints

La struttura del back-end del sistema e soprattutto dell'API è stata sviluppata tenendo in considerazione i principi dell'architettura **REST** che andrà a rispondere alle eventuali richieste HTTP in formato **JSON**.

L'API è stata sviluppata con un architettura **Front Controller** dove si accede ad essa tramite un singolo punto di accesso che inoltra la richiesta ad un "**Controller**" appropriato.

All'API si accede tramite un URI appropriato che descrive verbalmente la risorsa richiesta e/o l'azione che si vuole eseguire.

Un esempio di URI di BookExchange è il seguente:

<http://localhost:8080/bookexchange/api.php/book/list?q=science&limit=2>

Che fornirà come risposta l'elenco di libri trovati (in questo caso limitato a 2 elementi) che corrispondono alla query **q** formattati in **JSON** per esempio questa richiesta fornirà la seguente risposta:

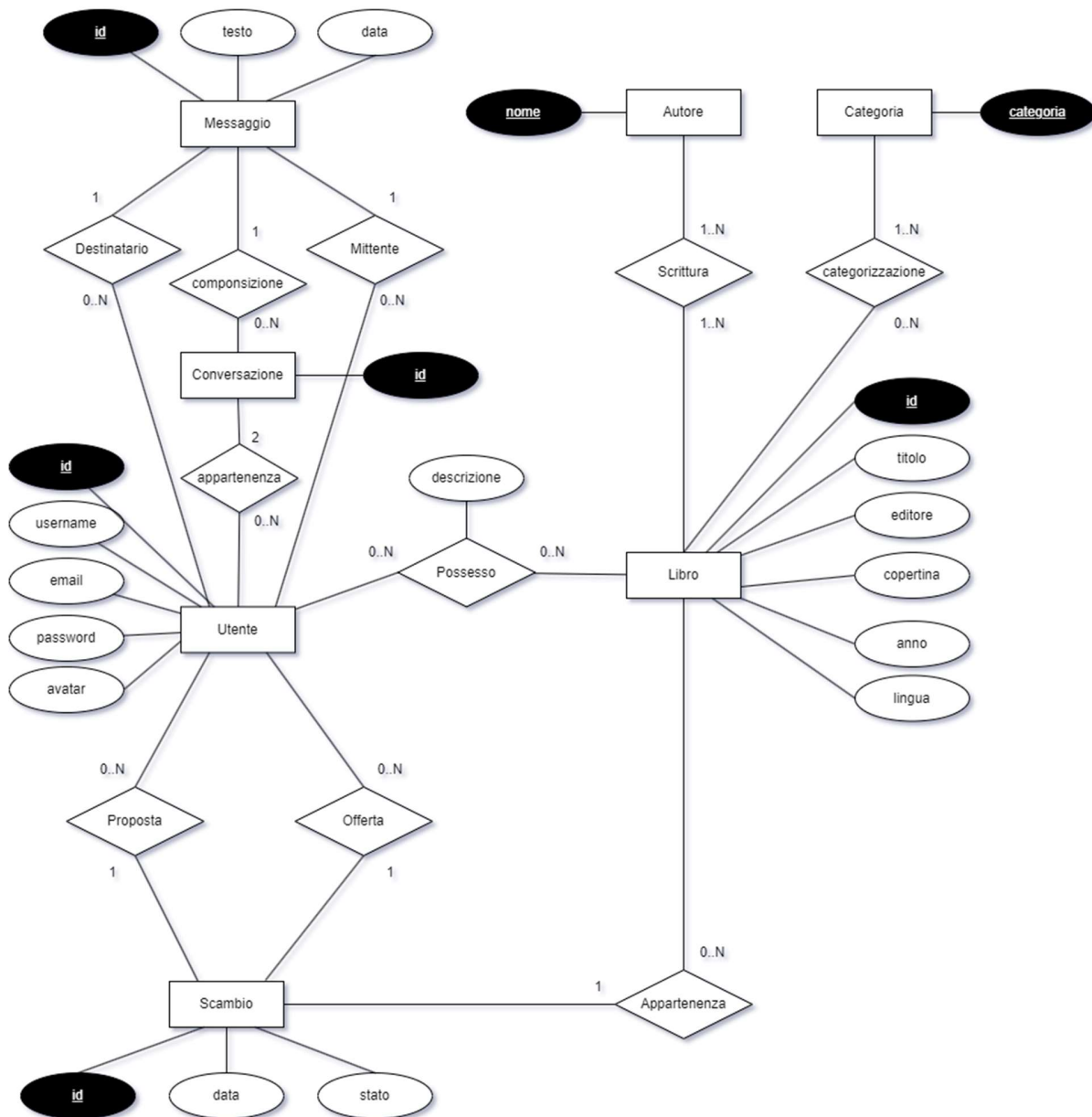
```
[
  {
    "id": "qR8DAAAAMBAJ",
    "volumeInfo": {
      "title": "Popular Science",
      "publishedDate": "1872-05",
      "imageLinks": {
        "smallThumbnail": "http://books.google.com/books/content?id=qR8DAAAAMBAJ&printsec=frontcover&img=1&zoom=5&edge=curl&source=gbs_api",
        "thumbnail": "http://books.google.com/books/content?id=qR8DAAAAMBAJ&printsec=frontcover&img=1&zoom=1&edge=curl&source=gbs_api"
      },
      "language": "en"
    }
  },
  ...
]
```

```
{
  "id": "HegDAAAAMBAJ",
  "volumeInfo": {
    "title": "Competition Science Vision",
    "publishedDate": "1999-06",
    "imageLinks": {
      "smallThumbnail": "http://books.google.com/books/content?id=HegDAAAAMBAJ&printsec=frontcover&img=1&zoom=5&edge=curl&source=gbs_api",
      "thumbnail": "http://books.google.com/books/content?id=HegDAAAAMBAJ&printsec=frontcover&img=1&zoom=1&edge=curl&source=gbs_api"
    },
    "language": "en"
  }
}
```


2.6 Diagrammi di progettazione

2.6.1 Modello ER – Progettazione concettuale

Di seguito è riportato il diagramma entità-relazione del sistema:



2.6.2 Progettazione logica del database

A partire dalla progettazione concettuale sono state individuate le seguenti relazioni che saranno le tabelle presenti all'interno del database relazionale scelto.

- Utente(**id**, username, email, password, avatar);
- Libro(**id**, titolo, editore, copertina, lingua, anno);
- Autore(**id**);
- Categoria(**categoria**, **libro**);
- Possesso(**proprietario**, **libro**, **descrizione**);
- Scrittura(**autore**, **libro**);
- Scambio(**id**, proponente, offerente, libro_proposto, libro_offerto, stato, data_creazione);
- Conversazione(**id**, utente1, utente2)
- Messaggio(**id**, conversazione, mittente, destinatario, data_creazione);

In cui sono definiti i vincoli di chiave, da cui poi si ricaveranno i vari vincoli inter-relazionali.

2.6.3 Dump del database

Di seguito è riportato il file contenente il “**Dump**” del database, che contiene tutte le istruzioni di creazione delle tabelle e dei vincoli progettati e inoltre contiene tutte le operazioni di inserimento dei dati presenti al suo interno.

3. External Interface Requirements

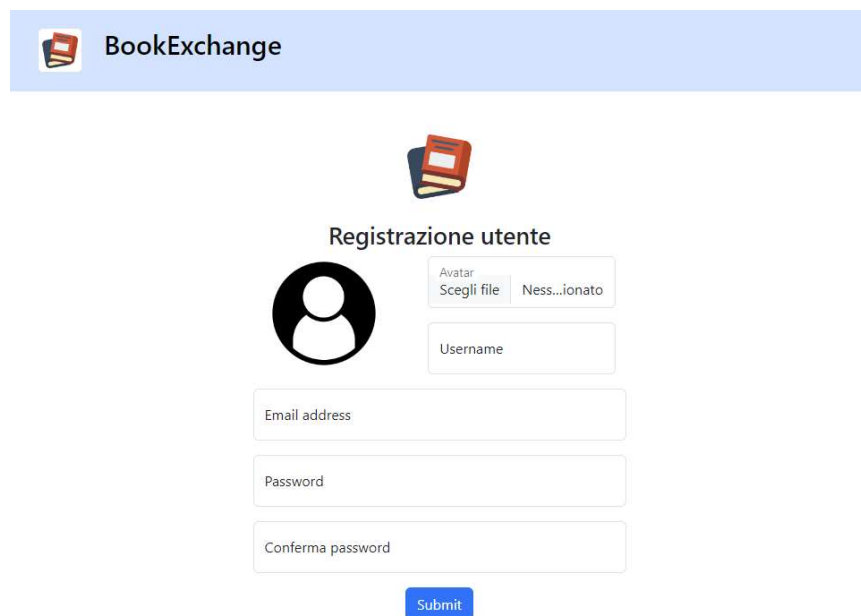
3.1 User Interfaces

3.1.1 Interfaccia grafica per l'utente o ospite del servizio

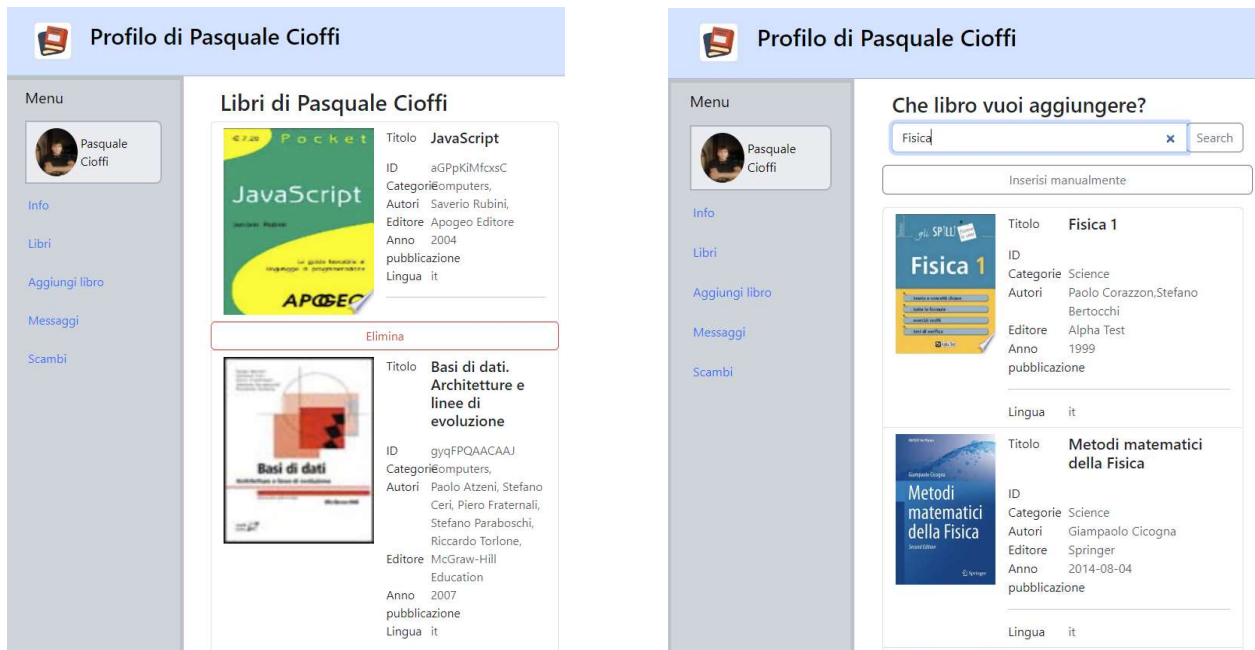
Per l'interfaccia utente è stato utilizzato un design minimale e intuitivo per l'utente inoltre grazie all'utilizzato di Bootstrap tutte le pagine fornite si adattano facilmente a qualsiasi tipo di display o dispositivo, alcuni esempi dell'interfaccia utente sono:



Questo è uno screenshot della home page dell'applicazione dove l'utente può accedere o registrarsi oppure tramite la barra di ricerca l'utente può inserire i dati del libro che sta cercando e verrà fornita una lista appropriata dei libri inseriti dagli altri utenti.



Qui invece è riportata la schermata di registrazione di un utente, nella quale l'utente può inserire le sue informazioni personali che saranno validate prima di effettuare la richiesta di registrazione al server.



Sopra sono riportate due schermate di esempio del profile dell'utente, la prima rappresenta la lista dei libri dell'utente, mentre la seconda è la schermata di ricerca per l'aggiunta di libri al proprio catalogo.

3.1.2 Interfaccia dell'API per un REST Client

Per accedere ai servizi forniti dall'API di BookExchange si parte dal singolo entry point:

<http://localhost:8080/bookexchange/api.php/>

A cui poi si possono alle varie risorse tramite URI e formattando la richiesta HTTP con i dati e tipo di richiesta appropriate, come da standard REST e per le richieste di tipo CRUD, per le richieste di lettura sono accettate richieste di tipo HTTP GET e gli eventuali dati sono inviati direttamente nell'URI, mentre per le richieste di scrittura/aggiornamento sono accettate richieste di tipo HTTP POST e i dati andranno inviati nel corpo della richiesta, non saranno quindi visibili nell'URI.

3.2 Software Interfaces

Il sistema opera tramite:

- **PHP 8.1.24** per le operazioni lato server.
- **10.4.28-MariaDB - mariadb.org binary distribution** per la gestione del database.
- **phpMyAdmin** per la gestione di richieste al web-server.
- **HTML 5** per la struttura delle pagine web.
- **JavaScript** per l'interazione con l'utente lato client.
- **Google Books API** per l'acquisizione di dati sui libri.
- **HTTP** per i protocolli di comunicazione.

3.3 Communications Interfaces

Il sistema elabora ogni richiesta tramite protocollo **HTTP** seguendo gli standard delle operazioni **CRUD** (Create Read Update Delete) e risponde allo stesso modo formattando i dati delle risposte in oggetti **JSON**.

Tutte le operazioni di lettura di informazioni del sistema richiedono richieste **HTTP GET** mentre ogni azione di modifica o inserimento di dati tramite richieste **HTTP POST**.

Di seguito sono riportate le risorse con i loro corrispondenti metodi e tipo di richieste accettate:

- **user** - [**http://localhost:8080/bookexchange/api.php/user:**](http://localhost:8080/bookexchange/api.php/user:)
 - **POST:**
 - /login** – permette all'utente di accedere al sistema, nel corpo della richiesta dovranno essere inviate la email e la password dell'utente. Verrà fornito come risposta un oggetto JSON che conterrà l'esito della operazione.
 - /register** – permette all'utente di registrarsi nel sistema, nel corpo della richiesta dovranno essere inviate la email, la password, lo username e l'avatar scelto dall'utente. Verrà fornito come risposta un oggetto JSON che conterrà l'esito della operazione.
 - /logout** – permette all'utente di eliminare la propria sessione e di disconnettersi dal sistema.
 - **GET:**
 - /list** – fornisce la lista degli utenti che sono registrati nel sistema.
 - / {userid}** – fornisce l'utente associato al seguente id.
 - / {userid} /profile** – fornisce la pagina del profile associato all'utente con id {userid}
 - / {userid} /books** – fornisce la lista di libri posseduti dall'utente con id {userid}.
- **book** - [**http://localhost:8080/bookexchange/api.php/book:**](http://localhost:8080/bookexchange/api.php/book:)
 - **POST:**
 - /add** – permette di inserire all'interno del catalogo dell'utente che effettua e la richiesta ed eventualmente nel database se non presente il libro le cui informazioni saranno riportate nel body della richiesta. Fornisce come risposta l'esito della operazione.
 - / {bookId} /delete** – permette di rimuovere dal catalogo dell'utente che effettua la richiesta il libro con id corrispondente a {bookId} i dati dell'utente proprietario vanno inseriti nel body della richiesta. Fornisce come risposta l'esito dell'operazione.
 - **GET:**
 - /search?q=&** – permette di ricercare tra i vari libri degli utenti quello con caratteristiche corrispondenti alla query **q**, fornisce come risposta una lista di libri con i giusti attributi inerenti alla ricerca.
 - /list?q=&limit** – permette di ricercare tramite l'API di GoogleBooks il libro con caratteristiche corrispondenti alla query **q**, fornisce come risposta una lista di libri con i giusti attributi inerenti alla ricerca.
 - / {bookId}** – fornisce come risposta il libro con id {bookId}
 - / {bookId} /authors** – fornisce come risposta la lista degli autori associate al libro con id {bookId}.
 - / {bookId} /categories** – fornisce come risposta la lista delle categorie associate al libro con id {bookId}.
 - / {bookId} /owners** – fornisce come risposta la lista degli utenti che hanno nel proprio catalogo il libro con id {bookId}.
- **message** - [**http://localhost:8080/bookexchange/api.php/message:**](http://localhost:8080/bookexchange/api.php/message:)
 - **POST:**

/startconv – permette di creare/inizializzare lo scambio di messaggi tra due utenti creando una nuova conversazione, nel body vanno riportati gli id dei due utenti interessati. Fornisce come risposta l'esito dell'operazione e in caso di successo la pagina web dove inviare o ricevere messaggi.

/send – permette di inviare un messaggio costituito da un blocco di testo tra due utenti (mittente e destinatario), nel body della richiesta vanno riportati gli id dei due utenti e il messaggio da inviare. Fornisce come risposta l'esito della operazione.

- **exchange** - <http://localhost:8080/bookexchange/api.php/exchange>:
 - **POST**:
 - /create** – permette di creare una richiesta di scambio di due libri tra due utenti, nel body vanno riportati l'id dell'utente offerente e proponente e l'id del libro offerto e quello proposto. Fornisce come risposta l'esito dell'operazione.
 - {exchangelid}** – permette di aggiornare lo stato della richiesta di scambio con id {exchangelid}, nel corpo della richiesta va specificato il nuovo stato, in caso di conferma il sistema effettuerà lo scambio dei libri associate alla scambio specificato.

4. System Features

4.1 Ricerca di libri tramite GoogleBooks

4.1.1 Description and Priority

Tramite l'apposito URI come richiesta GET e data una query di ricerca il sistema fornisce un elenco di libri con tutti gli attributi necessari alla memorizzazione sul Database (titolo, editore, autore/i, categoria/e, lingua, anno di pubblicazione).

Questa feature ha una priorità media in quanto agevole l'inserimento delle informazioni da parte dell'utente ma quest'ultime potrebbero anche essere inserite manualmente. Questa feature corre il rischio di non essere più adatta se il servizio fornito da Google dovesse cambiare il formato delle risposte.

4.1.2 Stimulus/Response Sequences

L'utente può accedere a questa feature accedendo al proprio profilo e navigando sulla sezione per l'inserimento di un libro

4.1.3 Functional Requirements

Per permettere all'**utente** di utilizzare questa feature, esso deve aver effettuato l'accesso al sistema, invece un attore del tipo **REST Client** può interrogare il sistema in un qualsiasi momento senza alcun tipo di autenticazione con l'apposita formattazione di richiesta **HTTP GET**

In caso di errore per esempio dei servizi di Google, Il sistema deve comunicare all'utente che non è stato possibile completare la ricerca.

REQ-1: l'utente deve aver effettuato l'accesso

REQ-2: il servizio di GoogleBooks deve essere online e funzionante

4.2 Ricerca di libri posseduti dagli utenti del sistema

4.2.1 Description and Priority

Permette agli utenti di ottenere, tramite richiesta GET con l'apposito URI oppure tramite la barra di ricerca della homepage, una lista di libri posseduti dagli altri utenti del sistema scelti in base alla query inviata.

4.2.2 Stimulus/Response Sequences

L'utente può accedere a questa feature tramite la barra di ricerca della homepage oppure tramite URI appropriato come richiesta GET.

4.2.3 Functional Requirements

REQ1: L'utente può filtrare la lista ricevuta in base a un query.

4.3 Scambio di libri tra utenti

4.3.1 Description and Priority

Gli utenti possono ricercare tra i libri disponibili per lo scambio quelli interessati e proporre al proprietario del libro uno scambio con un altro libro che l'utente possiede (ha aggiunto alla sua lista),

Questa feature ha la massima priorità in quanto è l'obiettivo principale della piattaforma quello dello scambio dei libri.

4.3.2 Stimulus/Response Sequences

L'utente può ricercare nella home page tra i libri disponibili quello che desidera supponendo che ci sia un utente che ne possieda uno.

Una volta trovato l'utente può selezionare il pulsante per proporre lo scambio, inserire il suo libro che vuole scambiare e inviare la proposta all'offerente.

L'utente potrà inoltre restare in contatto con il proprietario del libro tramite un sistema di messaggistica per coordinare il vero e proprio scambio.

In un qualsiasi momento entrambi gli utenti possono Annullare/Rifiutare/Accettare la richiesta di scambio.

4.3.3 Functional Requirements

Per poter utilizzare questa feature l'utente deve essere registrato nel sistema e deve aver effettuato l'accesso, inoltre deve disporre almeno di un libro nel suo catalogo.

In caso di errore nello scambio il sistema avviserà l'utente.

4.4 Scambio di messaggi tra utenti

4.4.1 Description and Priority

Gli utenti possono scambiarsi messaggi di testo tra una chat-box apposita

4.4.2 Stimulus/Response Sequences

L'utente potrà aprire delle chat (conversazioni) tra lui e qualsiasi altro utente della piattaforma, gli basterà navigare sul profilo dell'utente scelto e selezionare l'opzione "Message"

4.4.3 Functional Requirements

REQ1: l'utente deve essere registrato nel sistema e deve aver effettuato l'accesso.

4.5 Registrazione utente

4.5.1 Description and Priority

Permette all'utente di creare un suo profile/account sul sistema tramite l'inserimento di un email e password valida.

4.5.2 Stimulus/Response Sequences

L'utente selezione l'opzione di registrazione nella home page del sito, viene riportato a una pagina apposita dove può inserire le proprie informazione e inviarle, se queste sono valide viene inviata la richiesta di registrazione al sistema che se effettuata con successo creerà un nuovo utente con le informazioni inserite.

4.5.3 Functional Requirements

REQ-1: Le credenziali inserite dall'utente sono valide

4.6 Login utente

4.6.1 Description and Priority

Permette all'utente di accedere al sistema inserendo delle credenziali valide per un account già esistente.

4.6.2 Stimulus/Response Sequences

L'utente selezione l'opzione di login nella home page del sito, viene riportato a una pagina apposita dove può inserire le proprie informazione e inviarle, se queste sono valide viene inviata la richiesta di accesso al sistema che se avrà successo creerà una sessione che sarà valida fino al Logout dell'utente

4.6.3 Functional Requirements

REQ-1: Le credenziali inserite dall'utente sono valide
REQ-2: L'utente esiste nel database del Sistema

4.7 Messaggistica tra utenti

4.7.1 Description and Priority

Permette agli utenti registrati al Sistema di scambiarsi messaggi di testo

4.7.2 Stimulus/Response Sequences

L'utente naviga sul profilo dell'utente con la quale vuole avviare una conversazione, seleziona il pulsante appropriato e viene reindirizzato alla pagina della chat con l'utente scelto. Qui può scrivere il messaggio da inviare e cliccare sull'apposito pulsante per inviare il messaggio.

4.7.3 Functional Requirements

REQ-1: L'utente è registrato e ha effettuato l'accesso al sistema

REQ-2: L'utente può in qualsiasi momento accedere alle conversazioni con gli altri utenti.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Il sistema deve essere reattivo e veloce, deve fare uso di query efficienti che non vadano a rallentare le elaborazioni delle richieste.

5.2 Security Requirements

Il sistema non deve permettere di accedere a informazioni personali degli utenti, mantendone la privacy.

5.3 Software Quality Attributes

La struttura del software deve essere modulare e manutenibile in modo da poter effettuare facilmente modifiche e aggiornamenti. Inoltre l'API deve essere progettata in modo tale da essere utilizzabile anche da altre applicazioni.