



## ***Documento delle specifiche***

*Natalia Busuioc  
Terranova Luca*

## Sommario

<b>1. Introduzione.....</b>	<b>3</b>
1.1 Scopo del Documento.....	3
1.2 Descrizione del Progetto.....	3
1.3 Target Utenti.....	3
<b>2. Requisiti Funzionali.....</b>	<b>3</b>
RF1 - Gestione Utente.....	3
RF2 - Ricerca e Scoperta Libri.....	4
RF3 - Dettagli Libro.....	4
RF4 - Gestione Liste.....	4
<b>3. Requisiti Non Funzionali.....</b>	<b>4</b>
RNF1 - Usabilità.....	4
RNF2 - Performance.....	4
RNF3 - Sicurezza.....	5
RNF4 - Affidabilità.....	5
RNF5 - Scalabilità.....	5
<b>4. Architettura del Sistema.....</b>	<b>5</b>
4.1 Stack Tecnologico.....	5
4.2 Componenti Principali.....	5
4.3 Struttura e funzionamento di sistema.....	6
4.4 End Points.....	6
<b>5. Struttura Database.....</b>	<b>6</b>
Documentazione DbDocs: <a href="https://dbdocs.io/nataliabusioc27/SBL-Mobile">https://dbdocs.io/nataliabusioc27/SBL-Mobile</a> .....	6
5.1 Entità Principali.....	6
5.2 Entità Relazionali.....	6
5.3 Diagramma ER.....	7
5.4 Specifiche Tabelle.....	7
<b>6. Interfacce Utente - Schermate.....</b>	<b>8</b>
6.1 Login/Register.....	8
6.2 Home.....	8
6.3 Ricerca Libri.....	9
6.4 Focus Libro Selezionato.....	9
6.5 Profilo.....	9
6.6 Sezione Scaffali.....	10
6.7 Visualizzazione Scaffale Specifico.....	10
<b>7. Flussi Utente Principali.....</b>	<b>11</b>
7.1 Flusso Registrazione e Primo Accesso.....	11
7.2 Flusso Ricerca e Salvataggio Libro.....	11
7.3 Flusso Gestione Scaffali.....	11
<b>8. Integrazioni e Dati Esterni.....</b>	<b>11</b>
8.1 API per Dati Libri.....	11

# ==== **SBL** Mobile ====

8.2 Uso API esterne.....	12
<b>9. Roadmap e Fasi di Sviluppo.....</b>	<b>12</b>
Fase 1 - MVP (Minimum Viable Product).....	12
Fase 2 - Miglioramenti a breve termine.....	12
Fase 3 - Features Avanzate per sviluppi futuri.....	13
<b>10. Assegnazione Task.....</b>	<b>13</b>
<b>11. Note e Riferimenti.....</b>	<b>13</b>
Ispirazione Design.....	13
Tools.....	14
<b>12. Appendice.....</b>	<b>14</b>
12.1 Riferimenti.....	14

# Specifiche Tecniche

## 1. Introduzione

### 1.1 Scopo del Documento

Questo documento definisce le specifiche funzionali e tecniche per lo sviluppo di Simple Book Library (SBL), un'applicazione Android dedicata alla gestione e scoperta di libri.

### 1.2 Descrizione del Progetto

**SBL** (Simple Book Library) è un'applicazione Android che permette agli utenti di:

- Cercare informazioni dettagliate su libri specifici.
- Scoprire libri in trend e best seller.
- Creare e gestire liste personalizzate di lettura, previa registrazione e login.
- Tenere traccia dei propri progressi di lettura, previa registrazione e login.

Attraverso la creazione di un profilo personale, ogni utente può organizzare la propria libreria virtuale, salvando libri preferiti, libri da leggere, già letti e creare altre liste o categorie personalizzate da mostrare nel proprio profilo.

### 1.3 Target Utenti

- Utenti di ogni genere ed età, nuovi o già esperti lettori.
  - Lettori appassionati che vogliono organizzare le proprie letture.
  - Persone in cerca di nuovi libri da leggere.
  - Utenti che desiderano tenere traccia delle proprie statistiche di lettura.
- 

## 2. Requisiti Funzionali

### RF1 - Gestione Utente

- **RF1.1:** L'utente deve poter registrare un nuovo account fornendo: username e password
- **RF1.2:** L'utente deve poter effettuare il login con username e password
- **RF1.3:** L'utente deve poter visualizzare il proprio profilo e aggiungere, rimuovere o modificare le liste presenti in esso.
- **RF1.4:** L'utente deve poter visualizzare le proprie statistiche di lettura (numero libri letti, distribuzione per genere)

## RF2 - Ricerca e Scoperta Libri

- **RF2.1:** Il sistema deve mostrare una lista di libri bestseller/trending nella homepage anche senza aver effettuato registrazione o login.
- **RF2.2:** L'utente deve poter cercare libri tramite una barra di ricerca.
- **RF2.3:** I risultati della ricerca devono essere visualizzati in formato griglia.
- **RF2.4:** L'utente deve poter selezionare un libro per visualizzarne i dettagli completi in una schermata separata.

## RF3 - Dettagli Libro

- **RF3.1:** Il sistema deve mostrare le informazioni dettagliate di un libro selezionato, inclusi:
  - Titolo
  - Autore/i
  - Anno di pubblicazione
  - Immagine Copertina
  - Trama/descrizione
  - Genere
  - ISBN

## RF4 - Gestione Liste

- **RF4.1:** L'utente deve poter creare liste personalizzate con nome personalizzato.
  - **RF4.2:** L'utente deve poter aggiungere libri a una o più liste.
  - **RF4.3:** L'utente deve poter rimuovere libri dalle liste.
  - **RF4.4:** L'utente deve poter visualizzare tutti i libri contenuti in uno scaffale specifico.
  - **RF4.5:** Il sistema deve fornire liste predefinite suggerite (es. "Da leggere", "Letti", "Preferiti").
  - **RF4.6:** L'utente deve poter eliminare liste personalizzate.
- 

## 3. Requisiti Non Funzionali

### RNF1 - Usabilità

- L'interfaccia deve essere intuitiva e responsive per dispositivi mobile Android, con minima versione Android 13.0 ("Tiramisu").
- Il tempo di apprendimento per le funzioni base non deve superare i 10 minuti.

### RNF2 - Performance

- Il tempo di caricamento della homepage non dovrebbe superare i 2 secondi.
- I risultati di ricerca dovrebbero essere visualizzati entro 1 secondo dall'invio della query.

## **RNF3 - Sicurezza**

- Le chiamate API avverranno esclusivamente attraverso protocollo HTTPS.
- Le password devono essere crittografate nel database.
- L'accesso alle funzionalità di gestione liste deve essere limitato agli utenti autenticati.

## **RNF4 - Affidabilità**

- Il sistema deve garantire un uptime del 99%.
- I dati utente devono essere salvati in modo persistente.

## **RNF5 - Scalabilità**

- Il database deve essere progettato per gestire una crescita nel numero di utenti e libri.
- 

## **4. Architettura del Sistema**

### **4.1 Stack Tecnologico**

- **Frontend:** Android SDK (Java + XML).
- **Backend:** server JS usando Node.js + Express, ORM Sequelize (dialect: MySQL).
- **Database:** MySQL.
- **API Esterne:** Google Books API / Open Library API per i dati sui libri / NYT Books Api.

### **4.2 Componenti Principali**

1. **Modulo Autenticazione:** gestione login/registrazione.
2. **Modulo Ricerca:** interfaccia con API esterne per recupero dati dei libri.
3. **Modulo Gestione Liste:** CRUD operations sulle liste utente.
4. **Modulo Profilo:** visualizzazione e modifica dati utente.

## 4.3 Struttura e funzionamento di sistema

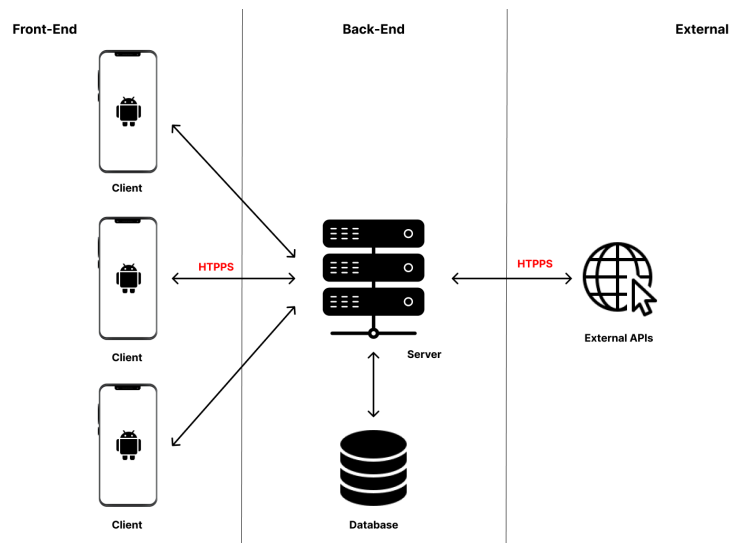
Il sistema ha una struttura classica e molto semplice:

- Client Front-end
- Struttura Back-End suddiviso in Server e Database

Il Server si occupa principalmente della distribuzione di dati ai clients richiedenti, oltre a questo si dedica all'ottenimento di tali dati eseguendo richieste alle API necessarie e, per migliorare le performance, salvandoli in cache temporaneamente per un lasso complessivo di ventiquattro ore.

Il Client richiede i dati al Server, solo quando necessario, attraverso protocollo HTTPS agli endpoints esposti.

Il Database si occupa del salvataggio delle informazioni riguardanti gli Utenti (user) e delle Librerie(shelf).



## 4.4 End Points (“/api”)

“/health”	
<i>Percorso</i>	“api/health”
<i>Metodo</i>	GET
<i>Descrizione</i>	permette di controllare se il server riceve e risponde correttamente alle chiamate
<i>Autenticazione</i>	Non richiesta
<i>Corpo della richiesta</i>	NON RICHIESTO
<i>Struttura della risposta</i>	<pre> JSON {   status: "Server is up and runnin" } </pre>
<i>Codici di stato</i>	- 200: success

“/register”	
<i>Percorso</i>	“api/register”
<i>Metodo</i>	POST
<i>Descrizione</i>	Permette di registrare una nuova utenza nel sistema, aggiungendo nome utente univoco e password criptata nel DB
<i>Autenticazione</i>	Non richiesta



# ==== **SBL Mobile** =====

<i>Corpo della richiesta</i>	<pre>JSON {   userName: "UtenteEsempio1",   password: "Password1234" }</pre>
<i>Struttura della risposta</i>	<pre>JSON {   message: "User profile created successfully",   userInfo: {     id: "123456",     userName: "UtenteEsempio1"   } }</pre>
<i>Codici di stato</i>	<ul style="list-style-type: none"> <li>- 200: success</li> <li>- 400: "Username and password are required"</li> <li>- 409: "An user with this name already exist"</li> <li>- 500: "There has been an internal server error"</li> <li>- 520: "Generic unknown error"</li> </ul>

<b>"/login"</b>	
<i>Percorso</i>	"api/login"
<i>Metodo</i>	POST
<i>Descrizione</i>	Permette di accedere al sistema trasmettendo nome utente e password, garantendo l'accesso all'applicazione se l'utente è già registrato
<i>Autenticazione</i>	Non richiesta

# ==== **SBL Mobile** =====

<i>Corpo della richiesta</i>	<pre>JSON {   userName: "UtenteEsempio1",   password: "Password1234" }</pre>
<i>Struttura della risposta</i>	<pre>JSON {   message: "User logged in sucessfully",   userInfo: {     id: "123456",     userName: "UtenteEsempio1"   } }</pre>
<i>Codici di stato</i>	<ul style="list-style-type: none"> <li>- 200: success</li> <li>- 401: "Wrong password"</li> <li>- 404: "User not found"</li> <li>- 500: "There has been an internal server error"</li> <li>- 520: "Generic unknown error"</li> </ul>

<b>"/getTrending"</b>	
<i>Percorso</i>	"api/getTrending"
<i>Metodo</i>	GET
<i>Descrizione</i>	Permette di ricevere 3 liste contenenti i libri bestsellers ottenuti dalla API di NYT Books
<i>Autenticazione</i>	Non richiesta
<i>Corpo della richiesta</i>	NON RICHiesto

# ==== **SBL Mobile** ====

<i>Struttura della risposta</i>	<pre>JSON {   bestsellers_date: "2025-12-11",   fiction: {     {       genre: "Fiction",       title: "Book Title",       author: "A person",       description: "Generic example book",       publication_year: "2023",       isbn: "9845623450123",       cover_url: "https://imagebook.it/sbdbehbak",     },     {       .....     },   }   nonfiction: {     {       genre: "Fiction",       title: "Book Title",       author: "A person",       description: "Generic example book",       publication_year: "2023",       isbn: "9845623450123",       cover_url: "https://imagebook.it/sbdbehbak",     },     {       .....     },   }   unifiedlist: {fiction + nonfiction} }</pre>
<i>Codici di stato</i>	- 200: success

**“/searchBook”**

# ==== **SBL Mobile** =====

<i>Percorso</i>	<p>"api/searchBook"</p> <p>Query param:</p> <ul style="list-style-type: none"> <li>- title: titolo del libro</li> <li>- author: autore del libro (opzionale)</li> <li>- isbn: codice isbn del libro (opzionale)</li> <li>- genere: genere del libro (opzionale)</li> </ul> <p>Esempio:</p> <p>"api/searchBook?title=Harry Potter&amp;genre=Fiction"</p>
<i>Metodo</i>	GET
<i>Descrizione</i>	permette di ricercare uno o più libri dal sistema che sono in qualche modo correlati ai parametri passati
<i>Autenticazione</i>	Non richiesta
<i>Corpo della richiesta</i>	NON RICHiesto
<i>Struttura della risposta</i>	<pre> JSON {   message: "succes",   books: {     {       genre: "Fiction",       title: "Book Title",       author: "A person",       description: "Generic example book",       publication_year: "2023",       isbn: "9845623450123",       cover_url: "https://imagebook.it/sbdbbehbak",     },     {       .....     },   } } </pre>
<i>Codici di stato</i>	- 200: success

## 5. Struttura Database

Documentazione DbDocs: <https://dbdocs.io/nataliabusioc27/SBL-Mobile>

Password: sblmobile

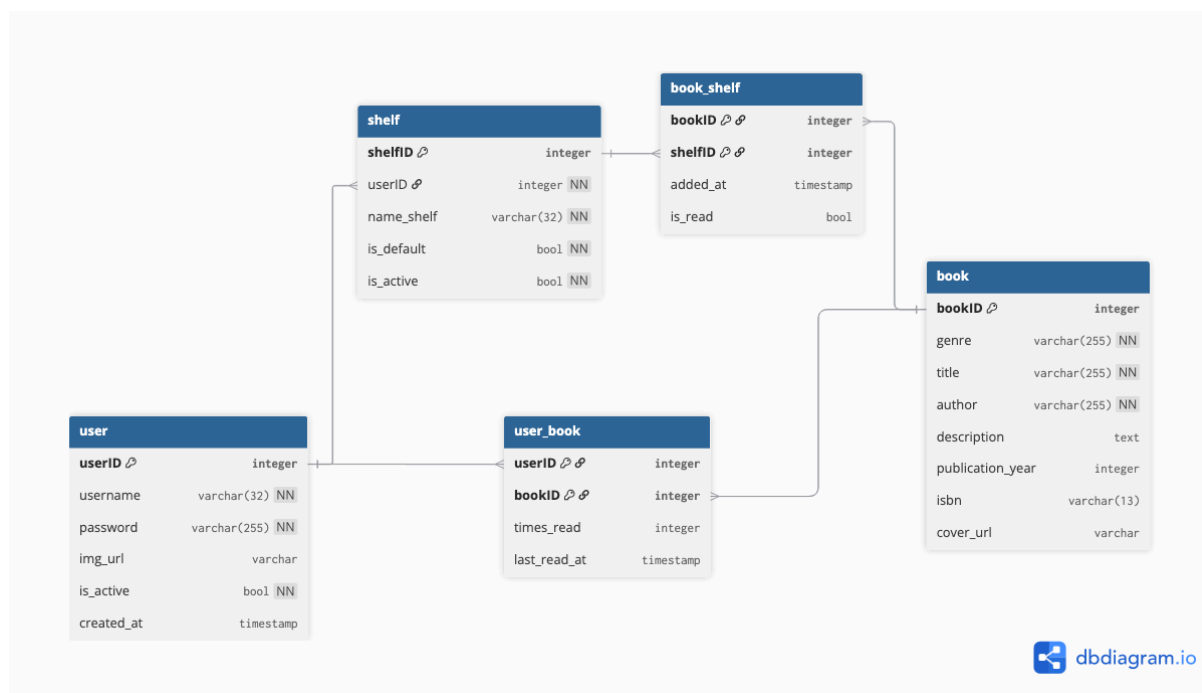
### 5.1 Entità Principali

- user
- shelf
- book

### 5.2 Entità Relazionali

- book\_shelf
- user\_book

### 5.3 Diagramma ER //nuova immagine



### 5.4 Specifiche Tabelle

user:

- userID - ID univoco del singolo user
- username - nome utente
- password - password criptata dell'account
- img\_url - immagine profilo utente

Versione documento: 5

Data ultima modifica: 2 Dicembre 2025

Autori: Luca Terranova, Natalia Busuioc

- is\_active - l'account è attivo o meno
- created\_at - quando l'account è stato creato

## **shelf:**

- shelfID - ID univoco del singolo scaffale
- userID - ID univoco dello user a cui appartiene lo scaffale (fa riferimento alla tabella **user**)
- name\_shelf - nome dello scaffale
- is\_default - flag che indica se lo scaffale è predefinito per ogni utente (Preferiti, Da leggere, Letti)
- Is\_active - flag che indica se lo scaffale è attivo o meno per l'utente (mostrato o nascosto )

## **book\_Shelf**

- bookID - ID univoco del singolo libro (fa riferimento alla tabella **book**)
- shelfID - ID univoco del singolo scaffale (fa riferimento alla tabella **shelf**)
- added\_At - data in cui il libro è stato aggiunto allo scaffale
- is\_read - flag che indica se il libro è stato letto

## **book**

- bookID - ID univoco del singolo libro
- genre - categoria a cui appartiene il libro
- title - titolo
- author - autore/i del libro
- description - trama del libro
- publication\_year - anno di pubblicazione
- isbn
- cover\_url - copertina del libro

## **user\_book**

- userID - ID dello user a cui appartiene il libro
- bookID - ID univoco del singolo libro
- times\_Read - numero di volte in cui il lettore ha letto lo stesso libro
- last\_Read\_At - indica la data dell'ultima volta che è stato letto il libro

---

## **6. Interfacce Utente - Schermate**

### **6.1 Login/Register**

#### **Funzionalità:**

- Form di registrazione con campi: username, password, conferma password.

- Form di login con campi: username, password.
- Link per passare da registrazione a login e viceversa.
- Validazione input lato client.

## Elementi UI:

- Input fields.
  - Pulsanti "Registrati" / "Accedi".
  - Link "Password dimenticata?" (opzionale per versioni future).
- 

## 6.2 Home

### Funzionalità:

- Visualizzazione libri trending/bestseller anche senza registrazione o login.
- Navigazione rapida alle altre sezioni.

### Elementi UI:

- Header con logo.
  - Sezione hero con call-to-action.
  - Griglia di libri in trend con copertine.
  - Footer con barra di navigazione rapida.
- 

## 6.3 Ricerca Libri

### Funzionalità:

- Barra di ricerca per inserire titolo/autore.
- Visualizzazione risultati in griglia.
- Click su libro per aprire dettagli.

### Elementi UI:

- Search bar prominente.
  - Griglia responsive di card libro.
  - Ogni card mostra: copertina, titolo, autore, breve descrizione, pulsante di salvataggio
  - Messaggio "Nessun risultato" se ricerca vuota.
- 

## 6.4 Focus Libro Selezionato

### Funzionalità:

- Visualizzazione dettagli completi del libro.
- Pulsante per salvare in uno scaffale.
- Possibilità di tornare ai risultati

## **Elementi UI:**

- Copertina grande.
  - Titolo, autore, anno, genere, ISBN.
  - Descrizione estesa.
  - Pulsante "Aggiungi allo scaffale" - Drop Down per selezionare lo scaffale di destinazione.
  - Pulsante "Indietro"
- 

## **6.5 Profilo**

### **Funzionalità:**

- Visualizzazione informazioni utente.
- Modifica dati profilo.
- Visualizzazione statistiche lettura.

### **Elementi UI:**

- Avatar/icona utente.
- Campi: username, generi preferiti.
- Sezione statistiche con:
  - Numero totale libri letti
  - Grafico distribuzione generi

### **Pop-up Statistiche:**

- Grafico a barre per generi
  - Contatori numerici
  - Pulsante chiusura
- 

## **6.6 Sezione Scaffali**

### **Funzionalità:**

- Visualizzazione tutti gli scaffali dell'utente.
- Creazione nuovo scaffale.
- Accesso a ciascun scaffale.

### **Elementi UI:**



- Griglia di card.
  - Ogni card mostra: nome scaffale, numero libri, anteprima copertine.
  - Pulsante "+ Nuovo Scaffale".
  - Pop-Up per creazione scaffale con input nome e descrizione.
- 

## 6.7 Visualizzazione Scaffale Specifico

### Funzionalità:

- Visualizzazione libri contenuti nello scaffale.
- Rimozione libri dallo scaffale.
- Possibilità di eliminare lo scaffale.

### Elementi UI:

- Header con nome scaffale.
  - Griglia di libri (copertina, titolo, autore).
  - Icona "X" su ogni libro per rimozione.
  - Pulsante "Elimina scaffale" (per liste non predefinite).
  - Messaggio "Scaffale vuoto" se non ci sono libri.
- 

## 7. Flussi Utente Principali

### 7.1 Flusso Registrazione e Primo Accesso

1. Utente accede alla pagina Login/Register.
2. Compila form di registrazione.
3. Viene creato account e reindirizzato alla Home.
4. Visualizza libri trending.
5. (Opzionale) Viene guidato alla creazione dei primi scaffali.

### 7.2 Flusso Ricerca e Salvataggio Libro

1. Utente dalla Home o menu accede a "Ricerca Libri".
2. Inserisce il termine di ricerca.
3. Visualizza i risultati in griglia.
4. Clicca su un libro interessante
5. Legge dettagli completi.
6. Clicca "Aggiungi allo scaffale".
7. Seleziona lo scaffale di destinazione (o ne crea una nuova).
8. Libro viene salvato nello scaffale.

### 7.3 Flusso Gestione Scaffali

Versione documento: 5

Data ultima modifica: 2 Dicembre 2025

Autori: Luca Terranova, Natalia Busuioc

1. Utente accede alla sezione "Scaffali" dal profilo.
  2. Visualizza tutte gli scaffali creati.
  3. Clicca su uno scaffale specifico.
  4. Visualizza i libri contenuti.
  5. Rimuove libri non più interessanti.
  6. Torna alla lista degli scaffali
- 

## 8. Integrazioni e Dati Esterni

### 8.1 API per Dati Libri

Opzioni considerate:

- **Google Books API:** dati completi, documentata, gratuita e completa con limiti generosi
- **Open Library API:** open source, dati estesi e completi, gratuita
- **NYT Books API:** open source, dati esterni, gratuita ma con limiti per il salvataggio dati, classifica settimanale bestsellers

Dati da recuperare:

- Classifica Bestsellers
- Informazioni bibliografiche
- Copertine
- Descrizioni

### 8.2 Uso API esterne

Libri bestseller / trending:

Per ottenere la lista di libri best sellers e di tendenza, il sistema farà affidamento da dati esterni ottenuti dalla API "New York Times Books API", la quale restituisce due liste separate di libri Fiction e Nonfiction, ordinate in base alla popolarità.

Tali libri verranno mostrati nella homepage.

In conformità con il regolamento presente nella documentazione API di *NYT*, i dati *NON verranno salvati per più di 24h*: per cui il server eseguirà 1 richiesta giornaliera salvando i dati fino alla successiva.

Ricerca libri:

Una volta inserito l'input per la ricerca, verrà inviata una richiesta al server back end, che a sua volta interrogherà l'API di "*Google Books*" e/o "*Open Library*" ottenendo i dati che verranno poi inviati di conseguenza all'utente richiedente.

Per migliorare le performance, i dati ricevuti verranno salvati sul server fino a nuova richiesta, evitando delay dovuti alla chiamata API

## 9. Roadmap e Fasi di Sviluppo

### Fase 1 - MVP (Minimum Viable Product)

- Sistema autenticazione
- Ricerca base libri
- Gestione liste basilare
- Profilo utente base









### Fase 2 - Miglioramenti a breve termine

- Statistiche di lettura avanzate
- Filtri ricerca avanzati
- Arricchimento informazioni profilo
- Condivisione liste (opzionale)
- Statistiche di lettura base

### Fase 3 - Features Avanzate per sviluppi futuri

- Algoritmo di raccomandazioni
- Integrazione social
- Sistema notifiche
- Export liste
- Possibilità di inserire note
- Sistema recensioni personali

## 10. Assegnazione Task

 Responsabile	 Title	 Date	 Status
Luca Terranova Natalia Petrone...	<b>Redazione SRS</b>	 Date	In progress ▾
Luca Terranova Natalia Petrone...	<b>Database</b> - Architettura & Diagramma	 Date	In progress ▾
Luca Terranova Natalia Petrone...	<b>Server</b>	 Data	In progress ▾
Luca Terranova Natalia Petrone...	<b>Power Point</b>	 Data	Completed ▾

👤 Responsabile	T Title	📅 Date	👁 Status
Natalia Petrone...	<b>Wireframe</b> - Register/Login	📅 Date	Completed ▾
Luca Terranova	<b>Wireframe</b> - Homepage	📅 Date	Completed ▾
Luca Terranova	<b>Wireframe</b> - Ricerca libro	📅 Date	Completed ▾
Natalia Petrone...	<b>Wireframe</b> - Profilo utente	📅 Date	Completed ▾
Luca Terranova	<b>Wireframe</b> - Focus libro selezionato	📅 Date	Completed ▾
Natalia Petrone...	<b>Wireframe</b> - Sezione Scaffali	📅 Date	Completed ▾
Natalia Petrone...	<b>Wireframe</b> - Visualizzazione scaffali	📅 Date	Completed ▾
Natalia Petrone...	<b>Android</b>	📅 Data	In progress ▾

## 11. Note e Riferimenti

### Ispirazione Design

- **TV Time:** applicazione di riferimento per tracking di serie TV/film/anime
- Adattare il concept al mondo dei libri con interfaccia pulita e intuitiva

### Tools

- **Wireframing:** Figma
- **Version Control:** Git/GitHub
- **DataBase structure / Docs:** dbDiagram + dbDocs
- **Development:** Android Studio / WebStorm / MySQL workbench

## 12. Appendice

### 12.1 Riferimenti

- Google Books API: <https://developers.google.com/books>
- Open Library API: <https://openlibrary.org/developers/api>

Versione documento: 5

Data ultima modifica: 2 Dicembre 2025

Autori: Luca Terranova, Natalia Busuioc

# ==== **SBL** Mobile ====

- NYT Books API: <https://developer.nytimes.com/docs/books-product/1/overview>
-