**Descrizione generale dell’applicazione**

L’esercitazione prevede la creazione di un software gestionale per una biblioteca.

L’applicazione permetterà di interfacciare l’utente con i contenuti della biblioteca, i quali saranno resi disponibili mediante una banca dati.

La versione finale dell’applicazione dovrà prevedere l’accesso per due tipi distinti di utenti: **utilizzatori** e **amministratori**.

Gli utenti utilizzatori dovranno essere in grado di accedere a una lista di libri disponibili e richiederne il prestito, nonché di formalizzarne la restituzione in qualsiasi momento rendendo di nuovo disponibile il titolo. Agli amministratori dell’applicazione sarà permesso tutto ciò che è permesso agli utilizzatori, con in più la possibilità di manipolare la banca dati, aggiungendo libri alla stessa, rimuovendone (a patto che non siano in prestito) o modificandone i dettagli.Nonostante sia previsto che la biblioteca possa disporre di più copie dello stesso libro, si richiede che un dato utente possa avere contemporaneamente in prestito un numero qualsiasi di libri, ma mai due copie dello stesso.   
  
Per semplicità, la singola richiesta di prestito è riferita ad un solo libro (in sostanza non dev’essere possibile prendere in prestito con la stessa richiesta due o più libri).

**Creazione entità e sviluppo di una Console Application**

L’applicazione dovrà da subito essere strutturata in maniera tale da separare lo strato di accesso ai dati dal “core” applicativo (business logic) e la GUI.

**Entità**  
Di seguito le entità fondamentali dell’applicazione:

* ***User*** rappresenta l’utente del sistema, ed è caratterizzato da:
  + ***UserId*** numero intero univoco
  + ***Username*** stringa
  + ***Password*** stringa
  + ***Role***, rappresentante il tipo di utenza (tipo consigliato: Enum)

I Role previsti sono due: Amministratore ed Utilizzatore

* ***Book*** rappresenta il libro in biblioteca ed caratterizzato da:
  + ***BookId*** numero intero univoco
  + ***Title*** stringa
  + ***AuthorName*** stringa
  + ***AuthorSurname*** stringa
  + ***PublishingHouse*** stringa
  + ***Quantity*** intero positivo (**NB.** Rappresenta solamente il numero globale di copie in biblioteca, NON il numero di copie disponibili al netto delle prenotazioni attive)
* ***Reservation*** rappresenta la prenotazione di un libro. E’ un’associativa tra libro ed utente, ed è caratterizzata da:
  + ***Id*** numero intero univoco
  + ***UserId*** che ha richiesto quella prenotazione
  + ***BookId*** richiesto
  + ***StartDate*** data che identifica l’inizio della prenotazione
  + ***EndDate*** data che identifica la fine della prenotazione (\*)

**NB**: Il libro è da considerarsi non più disponibile ad altre prenotazioni nel momento in cui esiste un numero di prenotazioni uguale alla quantità di copie in biblioteca aventi tutte un EndDate maggiore rispetto al momento in cui si fa richiesta. All’atto della restituzione, il sistema dovrà farsi carico di aggiornare l’EndDate della prenotazione con la data della restituzione stessa

(\*) L’EndDate deve essere valorizzata nel momento in cui viene effettuata la prenotazione come StartDate + 30 GG e aggiornata all’atto della restituzione del libro.

**Business Logic**  
  
Le funzionalità necessarie al corretto funzionamento del sistema, e dunque da implementare a livello Business Logic, sono le seguenti:

* ***Login:*** Dovrà essere valutata l’esistenza o meno dello User sulla base dei dati specificati dall’utente (Username, Password)
* ***Inserimento di un libro (solo admin):*** gestione inserimento di un libro (ovvero i dettagli anagrafici e quantità).   
  Va prestata attenzione nel caso in cui un libro sia già presente a sistema (\*) : in questo caso sarà necessario aggiornare la quantità di quello già presente e non inserirne uno nuovo.
* ***Modifica di un libro (solo admin):*** update dei dati anagrafici del Libro. NON è possibile modificare la quantità del libro, ma solamente i dati Anagrafici. Inoltre dovrà essere restituito un messaggio di errore nel caso in cui la modifica di un libro vada a causare un duplicato nel sistema (\*)
* ***Cancellazione di un libro (solo admin):*** Deve essere possibile cancellare un libro dal sistema se NON ci sono prenotazioni attive.   
  In caso vi siano una o più prenotazioni attive occorre notificare l’utente con un messaggio d’errore bloccante da mostrare sulla console dove vengono indicate le informazioni su chi ha il libro prenotato e fino a quando:

*“La cancellazione non è stata effettuata in quanto il libro XXXXX risulta essere ancora prenotato dall’utente YYYY a partire dal GG/MM/AAAA sino al GG/MM/AAAA”*

Tale messaggio dovrà essere mostrato tante volte quante sono le reservation attive del libro.

Se invece non vi sono prenotazioni attive, la cancellazione del libro deve prevedere anche la cancellazione delle sue eventuali prenotazioni passate

**NB:** Tale funzionalità permette di cancellare UN SOLO libro per volta. Non sono previste cancellazioni massive, per cui va gestita l’univocità del libro in fase di ricerca dello stesso.

* ***Ricerca di un libro:*** Sarà possibile ricercare uno o più libri filtrando la ricerca per uno o più parametri del libro (Titolo e/o Autore Nome e/o Autore Cognome e/o Casa Editrice).

E’ possibile ricercare tutti i libri non specificando nessun parametro di ricerca.

I risultati della ricerca dovranno mostrare l’Anagrafica del libro e se il libro sia prenotabile o meno (in ogni caso, dovrà essere mostrata la data in cui sarà disponibile la prenotazione del libro).

* ***Prestito di un libro***: Sarà necessario tenere conto dell’eventuale disponibilità di un libro: nel caso in cui **NON** sia disponibile, sarà necessario informare l’utente con un messaggio di errore (“*La prenotazione non è andata a buon fine in quanto il libro XXXXX risulta essere ancora prenotato* sino al *GG/MM/AAAA”*). La durata di default della prenotazione sarà di 30 giorni.

***Non è possibile effettuare un prestito per conto di un altro utente.***

Un utente NON può richiedere il prestito di un libro di cui ha già una prenotazione attiva.

* ***Restituzione del libro***: un utente può formalizzare la restituzione di un libro. Nel caso dovesse dichiarare la restituzione di un libro per il quale non sono memorizzate **sue** prenotazioni attive (EndDate maggiore della data di restituzione), il sistema dovrà restituire il seguente messaggio di errore: “*Il libro XXXXX non risulta essere attualmente in prestito.”*

***Non è possibile effettuare una restituzione per conto di un altro utente.***

Un libro risulta automaticamente restituito al termine dei 30 giorni.

* ***Visualizzare lo storico delle prenotazioni:***  questa ricerca può essere effettuata filtrando per Utente (Username) (solo admin) e/o Libro (Anagrafica) e/o Stato Prenotazione (Attiva/Non Attiva). Solo Admin può filtrare per Username, gli altri utenti devono filtrare per anagrafica del libro e stato di prenotazione.

Le informazioni da mostrare sono:

titolo del libro – nome utente – data di inizio prestito – data di fine prestito – info su stato prenotazione (attiva/non attiva)

* ***Uscita dall’applicativo***: l’utente può uscire dall’applicazione (chiusura console)

**Admin vs User**

Devono essere previste delle differenze su ciò che diversi tipi di utente possono fare, in base al ruolo.  
L’Amministratore ha accesso a tutte le funzionalità descritte in precedenza.

L’utente con ruolo utilizzatore **NON** ha accesso alle funzionalità:

***Inserimento di un libro  
Modifica di un libro  
Cancellazione di un libro***

Per le seguenti funzionalità invece sono previste delle limitazioni per l’utente con ruolo utilizzatore:

***Visualizzare lo storico delle prenotazioni:*** devono essere visualizzate solamente le informazioni relative alle proprie prenotazioni. In questo caso NON è quindi possibile filtrare la ricerca per utente.

(\*) Un libro si intende duplicato a sistema quando ha lo stesso titolo, stesso autore (Nome/Cognome) e stessa casa editrice.

**Layer di accesso ai dati (DAL)**

L’applicazione deve prevedere un livello interamente dedicato all’accesso dati (lettura/scrittura).  
  
Qualsiasi operazione di accesso ai dati deve dunque poter essere eseguita soltanto all’interno di questo layer: a tal fine, quest’ultimo deve esporre allo strato di Business Logic una serie di funzionalità di lettura e scrittura su tutte le entità in gioco.

Lo strato di accesso ai dati deve garantire totale trasparenza rispetto alla natura fisica della base dati.

Funzionalmente il DAL deve essere in grado di gestire le seguenti operazioni:

* Verificare i dati di login inseriti.
* Recupero di tutti i libri attualmente disponibili. (disponibili alla prenotazione o quantity > 0?)
* Recupero di un libro attraverso una, alcune, o tutte le sue proprietà.
* Inserimento di un libro.
* Aggiornamento di un libro esistente.
* Cancellazione di un libro.
* Recupero delle prenotazioni.
* Inserimento di una prenotazione.
* Cancellazione di una prenotazione.

In questa prima fase si richiede di implementare il DAL in maniera tale da gestire fisicamente la lettura/scrittura dei dati mediante utilizzo di file XML.

In particolare, i dati dovranno rispettare il seguente formato XML: 

Opzionalmente, è possibile mantenere tre file XML separati per i dati, uno per ogni diverso tipo di entità (books.xml, users.xml, reservations.xml).

Creare un DALRepository con le dependency injection di tutte le interfacce DAO

Creare IBookDao, IReservationDAO, IUserDao -> con le interfacce per i metodi CRUD

Creare BookDAO, ReservationDAO, UserDAO -> con i metodi CRUD

**Note Varie**

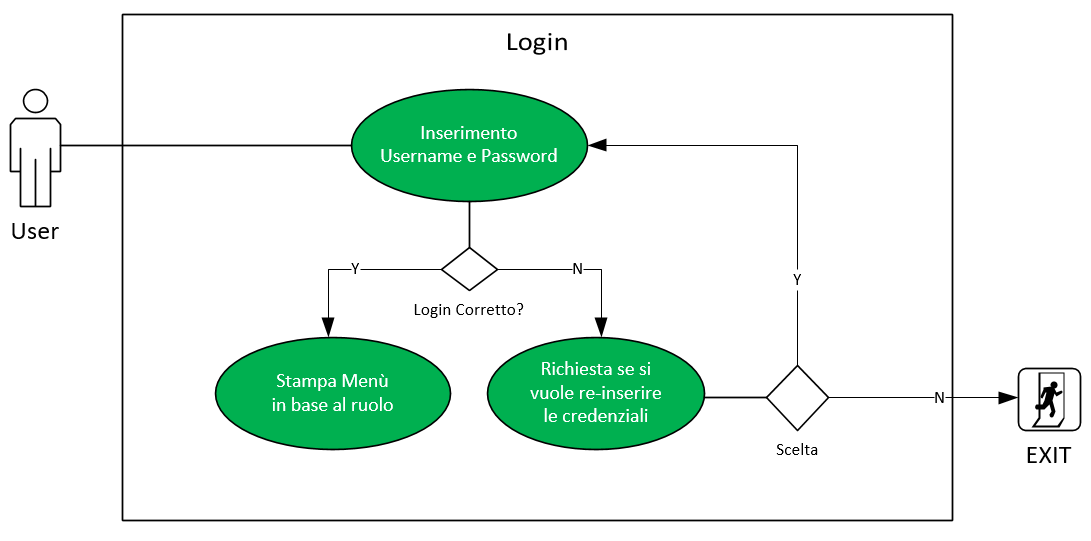
1. Login (UI): Deve essere possibile accedere all’applicativo solo dopo aver effettuato l’accesso con uno degli utenti presenti a sistema.  
   Se l’accesso non va a buon fine, l’utente può scegliere se uscire dall’applicativo.  
   Se l’accesso va a buon fine, l’utente può accedere alle funzionalità della biblioteca (in base al suo ruolo)
2. Per le seguenti funzionalità i parametri di ricerca del libro sono obbligatori e dunque devono essere sempre inseriti:
   1. Aggiunta di un libro (Anagrafica\* + Quantità)
   2. Modifica di un libro (Anagrafica\*) !!!
   3. Cancellazione di un libro (Anagrafica\*)
   4. Prenotazione di un libro (Anagrafica\*)
   5. Restituzione di un libro (Anagrafica\*)
   6. Storico Prenotazioni (nel caso in cui l’utente scelga di filtrare per libro) (Anagrafica\*)
3. **NON** è possibile effettuare ricerche per ID. Non è un’informazione nota all’Utente (Normale e/o Amministratore).
4. L’ID delle entità **NON** dovrà mai essere mostrato a video
5. E’ necessario rispettare il “*Separation of Concerns* *- SoC*” degli strati. In particolare:
   1. Le WriteLine e ReadLine devono essere presenti solamente in UI
   2. La logica di business (quella che non preveda la diretta interazione con l’Utente) dovrà essere presente solamente nella BusinessLogic
   3. La gestione di lettura e scrittura dei dati dovrà essere in carico solamente al DAL
6. I nomi dei metodi (CRUD) dei vari DAO devono essere coerenti rispetto a quello che fanno e rispetto allo strato in cui si trovano.  
   Ad esempio: La restituzione di un libro (azione di logica di business, NON di DAO!) viene tradotta come azione a DB come *update* di una *Reservation*, quindi il nome potrebbe essere “*Update”/”UpdateReservation”* e NON “*GiveBackBook*”/”*ReturnBook*”.

(\*) L’anagrafica del libro è rappresentata unicamente dai campi Titolo/Autore (Nome e Cognome)/Casa Editrice

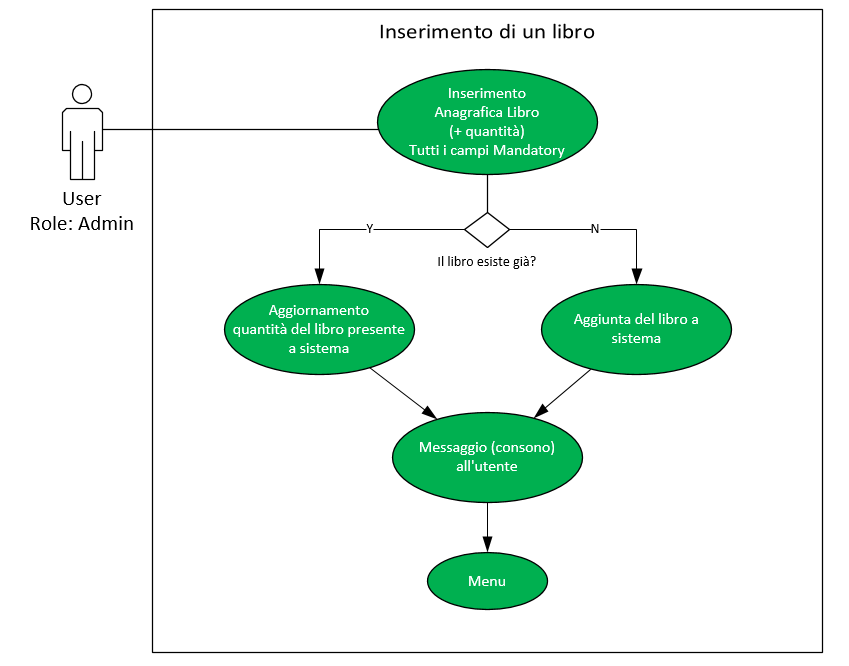
**Casi d’uso**

Di seguito sono presenti dei semplici casi d’uso e relativo *flow-chart* di **alcune** delle funzionalità del gestionale.

**Login**

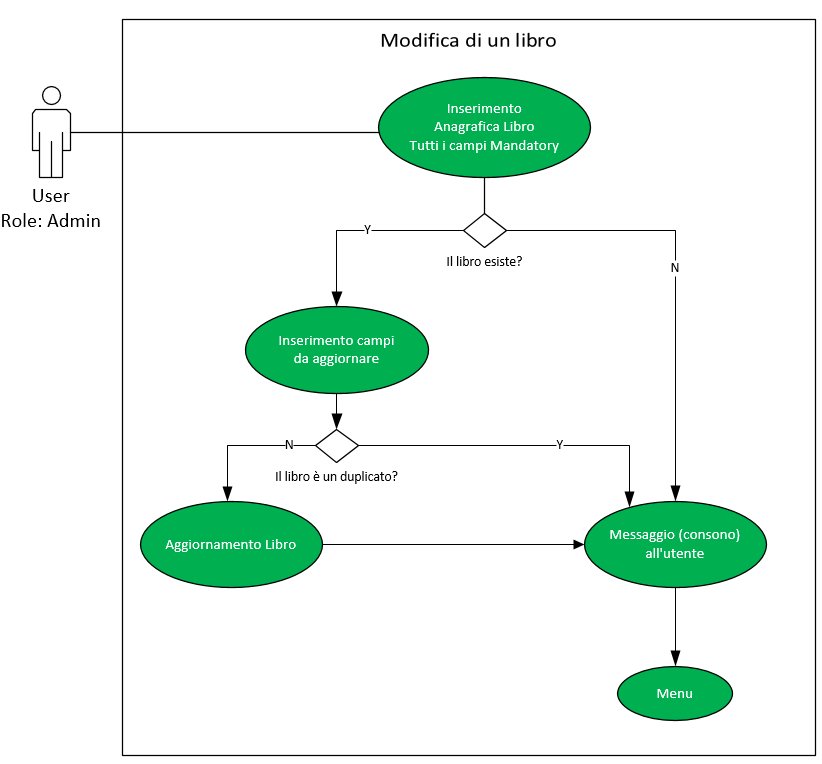


**Aggiunta di un libro**



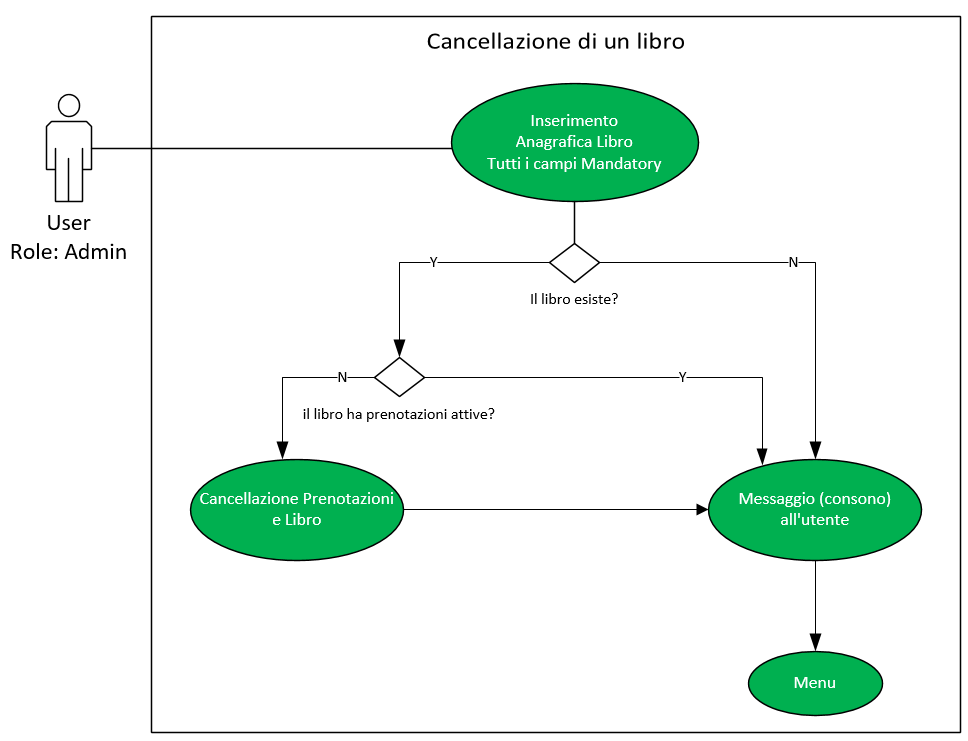
Mi manca il messaggio consono all’utente.

**Modifica di un libro**

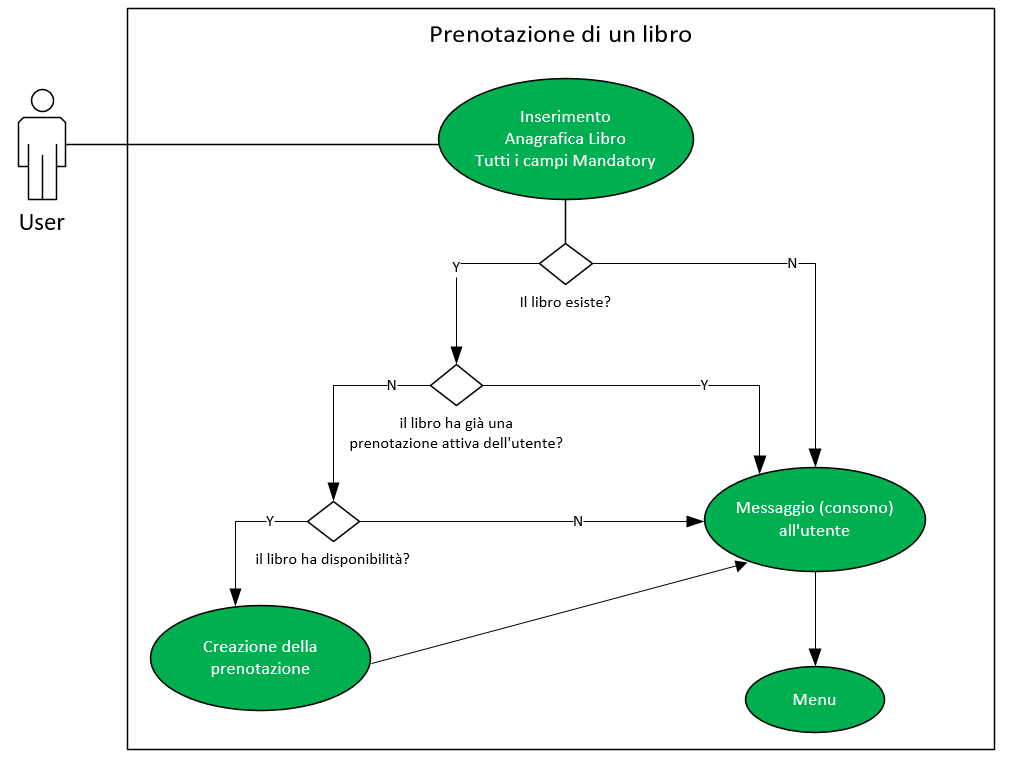


**duplicato???**

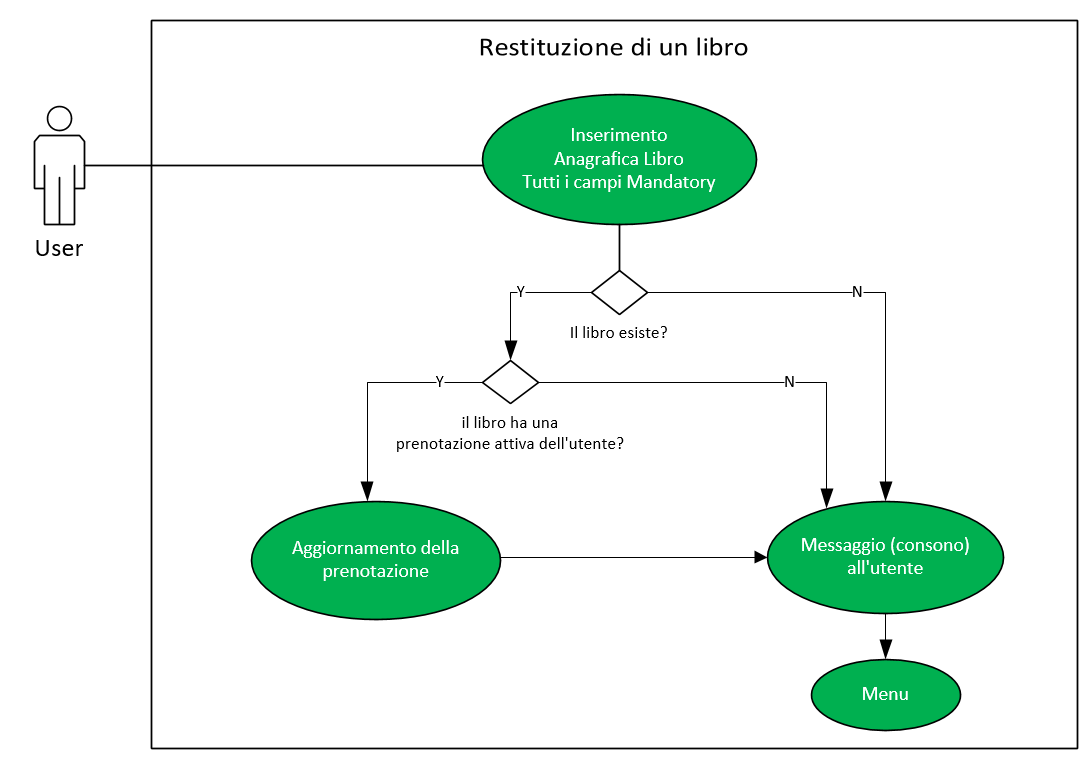
**Cancellazione di un libro**



**Prenotazione di un libro**



**Restituzione di un libro**



**Storico prenotazioni**

