

Documento di Specifica dei Requisiti

Progetto "Biblioteca Universitaria" (23)

Corso di Ingegneria del Software
Anno Accademico 2025/26



1. Descrizione dell'Applicazione

1. Gestione Libri
2. Gestione Utenti
3. Gestione Prestiti
4. Gestione autenticazione
5. Persistenza dei Dati

2. Requisiti Funzionali

Requisiti della categoria: Funzionalità Individuali

Business Flow

Requisiti della categoria: Requisiti relativi ai Dati

Requisiti della categoria: Requisiti dell'Interfaccia Utente

Tabella di Collegamento

Funzionalità Individuali ↔

Casi d'Uso

3. Requisiti Non Funzionali

Vincoli

4. Casi d'Uso

1. Gestione Libri

- UC-1 - Visualizzazione catalogo
- UC-2 – Inserimento Libro
- UC-3 – Modifica Libro
- UC-4 – Cancellazione Libro
- UC-5 – Ricerca Libro

2. Gestione Utenti

- UC-6 - Visualizzazione elenco utenti
- UC-7 – Inserimento Utente
- UC-8 – Modifica Utente
- UC-9 – Cancellazione Utente
- UC-10 – Ricerca Utente

3. Gestione Prestiti

UC-11 – Registrazione Prestito

UC-12 – Registrazione Restituzione

UC-13 – Visualizzazione Elenco Prestiti

UC-14 – Gestione Blacklist

4. Gestione Autenticazione

UC-15 – Login

UC-16 – Logout

UC-18 – Aggiornamento inventario

UC-19 – Aggiornamento database utenti

5. Diagrammi dei Casi d’Uso

1. Gestione Libri

2. Gestione Utenti

3. Gestione Prestiti

4. Gestione Autenticazione

6. Matrice di Tracciabilità

1. Descrizione dell'Applicazione

L'applicazione “Biblioteca Universitaria” supporta il bibliotecario nella gestione dei libri, degli utenti e dei prestiti in modo semplice e organizzato. Il sistema consente l'inserimento, la modifica, la rimozione e la ricerca dei dati, applicando automaticamente vincoli prestabiliti. Tutti i dati vengono salvati su file per garantire persistenza e continuità tra una sessione e l'altra. L'accesso alle funzionalità è protetto da un meccanismo di autenticazione con login e logout.

L'obiettivo è migliorare l'efficienza del lavoro del bibliotecario mettendo a disposizione un'interfaccia chiara e funzionale per svolgere rapidamente le operazioni quotidiane con una grafica “minimal” come se fosse un vero e proprio sistema per un bibliotecario.

Per rendere più chiara la struttura del sistema, le funzionalità sono organizzate in business flow, ciascuno contenente un gruppo specifico di operazioni:

1. Gestione Libri

Raccoglie tutte le funzionalità legate al catalogo: inserimento, modifica, eliminazione, ricerca(un tasto indietro) e visualizzazione dei libri.

2. Gestione Utenti

Comprende le operazioni relative agli utenti registrati: creazione, aggiornamento, eliminazione, consultazione dell'elenco e visualizzazione della blacklist con prestiti.

3. Gestione Prestiti

Si occupa dei flussi operativi per la registrazione dei prestiti e delle restituzioni, applicando automaticamente i vincoli della biblioteca.

4. Gestione autenticazione

Si occupa dell'accesso e uscita dal sistema e gestione delle credenziali del bibliotecario (login e logout).

5. Persistenza dei Dati

Include le operazioni di salvataggio e caricamento dei dati, necessarie per garantire che gli aggiornamenti non scompaiano.

2. Requisiti Funzionali

Requisiti della categoria: Funzionalità Individuali

Le funzionalità individuali descrivono i singoli servizi che il sistema deve offrire. Sono operazioni atomiche che l'utente può eseguire direttamente, come inserire o modificare un elemento.

Funzione	Ragione	Descrizione	Area	Priorità	Codice
Inserire un libro	Aggiornare catalogo	Inserimento di un nuovo libro in archivio	Gestione Libri	Must Have	FI-1
Modificare libro	Correggere/aggiornare info	Modifica dei dati registrati	Gestione Libri	Must Have	FI-2
Eliminare libro	Rimuovere volumi	Eliminazione dall'archivio se non in prestito	Gestione Libri	Must Have	FI-3
Ricercare libro	Trovare rapidamente	Ricerca per titolo, autore o codice	Gestione Libri	Must Have	FI-4
Inserire utente	Registrare utenti	Inserimento di un nuovo utente on database	Gestione Utenti	Must Have	FI-5
Modificare utente	Aggiornare info	Modifica dati utente registrato	Gestione Utenti	Should Have	FI-6
Eliminare utente	Gestire iscritti	Elimina utente senza prestiti attivi	Gestione Utenti	Should Have	FI-7
Ricercare utente	Trovare utente	Ricerca per matricola, cognome e nome	Gestione Utenti	Must Have	FI-8
Visualizzazione ordinata catalogo	Monitorare la situazione generale	Permette di mostrare in ordine di titolo i libri presenti nel	Gestione libri	Must have	FI-9

		sistema con le principali informazioni			
Registrare Prestito	Gestire formalmente la consegna dei libri	Registra un prestito inserendo matricola utente, isbn libro e data prevista di restituzione (max 3 mesi dal giorno del prestito)	Gestione Prestiti	Must have	FI-10
Registrare Restituzione	Aggiornare disponibilità copie e chiudere prestito	Registrazione della restituzione del libro prestato, aggiornamento disponibilità libri	Gestione Prestiti	Must have	FI-11
Controllo Disponibilità libri	Evitare prestiti quando non ci sono copie disponibili	Il sistema blocca la richiesta se le copie disponibili sono terminate	Gestione Prestiti	Must have	FI-12
Controllo limite prestiti per utente	Rispettare il regolamento interno (max 3 prestiti)	Il sistema blocca il prestito se l'utente ha già 3 prestiti ancora attivi	Gestione Prestiti	Must Have	FI-13
Visualizzare elenco prestiti	Monitorare la situazione generale dei prestiti	Permette di mostrare i prestiti attivi degli utenti, ordinati per data prevista di restituzione, evidenziando quelli in ritardo	Gestione Prestiti	Should Have	FI-14
Inserimento black list	Inserimento in una lista nera	Permette di inserire in lista nera un utente bloccando le operazioni di prestito.	Gestione Prestiti	Should Have	FI-15
Visualizzare utenti	Consultare elenco	Elenco utenti ordinato per cognome e nome	Gestione Utenti	Must Have	FI-16
Login	Accedere all'area riservata	Permette al bibliotecario di autenticarsi tramite password per accedere alle funzionalità riservate.	Gestione Utenti	Must Have	FI-17

Mascheramento password	Rendere il profilo più sicuro	Permette di migliorare la sicurezza delle credenziali, codificando la password	Gestione Utenti	Should Have	FI-18
Modifica Password	Modificare la password di accesso	Permette di modificare la password per migliorare ulteriormente la sicurezza del profilo	Gestione Utenti	Must Have	FI-19
Recupero Password	Password di accesso dimenticata	Permette di reimpostare la password nel caso in cui la precedente venga dimenticata	Gestione Utenti	Can Have	FI-20
Logout	Disconnessione dall'archivio	Permette di uscire dall'archivio, ritornando alla schermata di login	Gestione Utenti	Must Have	FI-21
Prenotazione libro	Gestire la richiesta di libri molto utilizzati	Permette al bibliotecario di registrare una prenotazione su un libro attualmente in prestito. Alla restituzione, il libro viene segnato come "prenotato" per l'utente indicato e non è disponibile per altri prestiti finché la prenotazione non viene evasa.	Gestione libri	Can Have	FI-22
Notifica automatica ritardi via e-mail	Ridurre i ritardi nella restituzione	Il sistema, integrato con un servizio esterno di posta elettronica, invia automaticamente una mail di promemoria agli utenti con prestiti scaduti, indicando i libri in ritardo e la nuova data entro cui restituirli.	Gestione Prestiti	Can Have	FI-23

Business Flow

Funzione	Ragione	Descrizione	Area	Priorità	Codice
Registrare prestito	Gestire formalmente un prestito	Flusso che comprende la selezione dell'utente (FI-8), la selezione del libro (FI-4), il controllo della disponibilità copie (FI-12, VC-2) e del limite massimo di prestiti per utente (FI-13, VC-1), fino alla registrazione del prestito (FI-10)	Gestione Prestiti	Must Have	BF-1
Registrare restituzione	Aggiornare lo stato dei prestiti	Flusso che comprende la selezione del prestito attivo(FI-14), la conferma della restituzione (FI-11) e l'aggiornamento delle copie disponibili (collegato a FI-12, VC-3)."	Gestione Prestiti	Must Have	BF-2
Gestione catalogo libri	Mantenere il catalogo aggiornato e consultabile	Flusso che comprende l'inserimento di nuovi libri (FI-1), la modifica dei dati di un libro (FI-2), l'eliminazione di libri (FI-3), la ricerca nel catalogo (FI-4) e la visualizzazione dell'elenco dei libri (FI-9)	Gestione Libri	Must Have	BF-3
Gestione Utenti Biblioteca	Gestire gli utenti registrati	Flusso che comprende l'inserimento di nuovi utenti (FI-5), la modifica dei dati utente (FI-6), l'eliminazione di utenti (FI-7) e la ricerca/visualizzazione di un utente registrato (FI-8/ FI-16)	Gestione Utenti	Must Have	BF-4
Gestione autenticazione	Proteggere l'accesso al sistema	Flusso che comprende l'accesso del bibliotecario al sistema tramite login (FI-17), con mascheramenti, modifica e recupero della password (FI-19, FI-18, FI-20) e il logout (FI-21).	Gestione Utenti / Autenticazione	Must Have	BF-5

Requisiti della categoria: Requisiti relativi ai Dati

Riguardano i dati che il sistema deve memorizzare, elaborare o mantenere. Descrivono obblighi e vincoli relativi alla persistenza e gestione delle informazioni.

Funzione	Ragione	Descrizione	Area	Priorità	Codice
Gestire dati libro	Memorizzare informazioni complete sui libri	Il sistema deve memorizzare per ogni libro titolo (stringa), lista degli autori (lista di stringhe), anno di pubblicazione (localDate), codice identificativo univoco (int) e numero di copie disponibili (int)	Gestione Libri	Must Have	DT-1
Gestire dati Utente	Memorizzare informazioni complete sugli utenti	Il sistema deve memorizzare per ogni utente almeno nome, cognome, e-mail istituzionale e matricola (stringhe), e la lista dei libri attualmente in prestito con la relativa data prevista di restituzione.	Gestione Prestiti e Utente	Must Have	DT-2
Archivio persistente	Garantire persistenza e coerenza dei dati	L'intero archivio (libri, utenti e prestiti) deve essere caricato da file all'avvio e salvato su file dopo le operazioni di inserimento, modifica o eliminazione, in modo da mantenere i dati aggiornati tra una sessione e l'altra	Persistenza dati	Must Have	DT-3
Statistiche sui prestiti	Fornire una visione sintetica sull'utilizzo della biblioteca	Schermata riepilogativa con semplici indicatori statistici: numero di prestiti per periodo, elenco dei libri più prestati, utenti più attivi e numero di prestiti in ritardo. Le statistiche	Gestione Prestiti	Can Have	DT-4

		sono solo in lettura e non modificano i dati gestionali.			
--	--	--	--	--	--

Requisiti della categoria: Requisiti dell'Interfaccia Utente

Descrivono come il sistema deve presentare le informazioni e supportare l'interazione dell'utente. Includono visualizzazione, usabilità e struttura dell'interfaccia.

Funzione	Ragione	Descrizione	Area	Priorità	Codice
Schermata di autenticazione	Proteggere l'accesso al software	Schermata visualizzata all'avvio del software con il campo per l'inserimento della password e il comando "Hai dimenticato password" con ulteriore schermata	Autenticazione	Must Have	UI-1
menu view	Scegliere rapidamente l'area di lavoro	Schermata iniziale con i comandi "Gestione Libri", "Gestione Utenti", "Gestione Prestiti", "Logout".	Persistenza dati	Must Have	UI-2
Schermata gestione / ricerca libri	Cercare e gestire i libri	Sezione "Libri" con tabella dei libri, campo di ricerca e pulsanti Inserisci Modifica Elimina e Cerca che diventa indietro	Gestione Utenti	Must Have	UI-3

Schermata gestione prestito / restituzione	Gestire prestiti, restituzioni e blacklist	Sezione Prestiti con tabella prestiti attivi (ordinati per data di restituzione), evidenza dei ritardi e pulsanti Nuovo prestito Restituzione Blacklist utente. Blocco dei prestiti per utenti in blacklist con messaggio di avviso.	Gestione Prestiti	Must Have	UI-4
Schermata gestione utenti	Registrare e aggiornare i dati degli utenti	Sezione “Utenti” con tabella utenti (incluso stato blacklist), campo di ricerca e pulsanti Inserisci Modifica Elimina Blacklist Cerca	Gestione Utenti	Must Have	UI-5
Tema grafico chiaro/scuro	Migliorare la fruibilità dell’interfaccia	L’applicazione offre un’impostazione per scegliere tra tema chiaro e tema scuro. La scelta viene applicata a tutte le schermate della GUI e mantenuta tra una sessione e l’altra.	Persistenza Dati	Can Have	UI-6

Tabella di Collegamento

Funzionalità Individuali ↔ Caso d'Uso

Codice FI	Funzionalità Individuale	Caso d'Uso Associato
FI-1	Inserire libro	UC-2 Inserimento libro
FI-2	Modificare libro	UC-3 Modifica libro
FI-3	Eliminare libro	UC-4 Cancellazione libro
FI-4	Ricercare libro	UC-5 Ricerca libro
FI-5	Inserire utente	UC-7 Inserimento utente
FI-6	Modificare utente	UC-8 Modifica utente
FI-7	Eliminare utente	UC-9 Cancellazione utente
FI-8	Ricercare utente	UC-10 Ricerca utente
FI-9	Visualizzazione Catalogo	UC-1 Visualizzazione catalogo
FI-10	Registrare Prestito	UC-11 Registrazione Prestito
FI-11	Registrare restituzione	UC-12 – Registrazione restituzione

FI-12	Controllare disponibilità copie libro	UC-1 – Visualizzazione catalogo
FI-13	Controllare limite prestiti per utente	UC-11 – Registrazione prestito
FI-14	Visualizzare elenco prestiti	UC-13 – Visualizzazione elenco prestiti
FI-15	Inserimento utente in blacklist	UC-14 – Gestione blacklist
FI-16	Visualizzare elenco utenti	UC-6 – Visualizzazione elenco utenti
FI-17	Login	UC-15 – Login
FI-18	Mascheramento Password	UC-15 – Login
FI-19	Modifica Password	UC-17 – Modifica password
FI-21	Logout	UC-16 – Logout

3. Requisiti Non Funzionali

Funzione	Ragione	Descrizione	Priorità	Codice
Usabilità	Interfaccia semplice	L'interfaccia deve essere chiara, intuitiva e con funzioni facilmente accessibili.	Should Have	NFR-1
Affidabilità	Evitare perdita dati	Il sistema deve prevenire errori e garantire che i dati salvati siano recuperabili.	Must Have	NFR-2
Prestazioni	Risposta rapida	Il sistema deve fornire risposte rapide senza ritardi evidenti.	Should Have	NFR-3
Manutenibilità	Facile intervento	Il codice deve essere strutturato per permettere modifiche senza effetti collaterali.	Must Have	NFR-4
Portabilità	Uso su sistemi diversi	L'applicazione deve poter funzionare su sistemi operativi differenti.	Should Have	NFR-5
Robustezza	Gestione errori	Il sistema deve gestire correttamente input errati, file mancanti e formati non validi.	Must Have	NFR-6

Sicurezza	Protezione dati	Il sistema deve evitare corruzioni o perdite dei dati sensibili.	Must Have	NFR-7
-----------	-----------------	--	-----------	-------

Vincoli

Funzione	Ragione	Descrizione	Priorità	Codice
Vincolo funzionale	Limite prestiti	Un utente non può avere più di tre prestiti attivi.	Must Have	VC-1
Vincolo funzionale	Disponibilità copie	Un libro può essere prestato solo se almeno una copia è disponibile.	Must Have	VC-2
Vincolo funzionale	Aggiornamento copie	La restituzione deve aggiornare automaticamente la disponibilità delle copie.	Must Have	VC-3
Vincolo funzionale	Eliminazione soggetti	Non si possono eliminare libri o utenti coinvolti in prestiti attivi.	Must Have	VC-4
Vincolo di Coerenza	ID libro univoco	Ogni libro deve possedere un identificativo univoco.	Must Have	VC-5

Vincolo di Coerenza	ID utente univoco	Ogni utente deve avere una matricola.	Must Have	VC-6
Vincolo di Coerenza	No duplicati	Non devono esistere elementi duplicati non intenzionali.	Must Have	VC-7
Vincolo di Coerenza	Prestiti validi	Ogni prestito deve riferirsi a utenti e libri registrati.	Must Have	VC-8
Vincolo Essenziale	Rispetto regolamento	Il sistema deve rispettare il regolamento interno della biblioteca.	Must Have	VC-9

4. Casi d'Uso

1. Gestione Libri

UC-1 - Visualizzazione catalogo

Codice: UC-1

Titolo: Visualizzazione catalogo

Attore partecipante: Bibliotecario

Precondizioni:

- Bibliotecario autenticato.
- Sistema correttamente avviato.

Postcondizioni:

- Il catalogo dei libri è visualizzato in forma tabellare ordinato per titolo.

Flusso di eventi:

1. Il bibliotecario seleziona "Gestione libri".
2. Il sistema carica dal file/archivio l'elenco dei libri presenti.
3. Il sistema mostra il catalogo ordinata per titolo in una tabella contenente codice ISBN, titolo, autore, anno e copie disponibili.

Flussi alternativi:

- 2a Nessun libro registrato → il sistema mostra una tabella vuota o un messaggio "Nessun libro presente in catalogo".

UC-2 – Inserimento Libro

Codice: UC-2

Titolo: Inserimento Libro

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema correttamente avviato.
- Il bibliotecario è autenticato e seleziona "Gestione Libri".

Postcondizioni:

- Il nuovo libro risulta registrato.

Flusso di eventi:

1. Il bibliotecario inserisce i dati del libro.
2. Seleziona 'Inserisci libro'.
3. Il sistema aggiorna l'inventario.

Flussi alternativi:

- 3a. Dati mancanti / formato incorretto → richiesta completamento / reinserimento

3b. Codice ISBN già presente in inventario → inserimento bloccato e messaggio di notifica

UC-3 – Modifica Libro

Codice: UC-3

Titolo: Modifica Libro

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema correttamente avviato.
- Il bibliotecario è autenticato e seleziona “Gestione libri”.
- Almeno un libro esistente nell’inventario.

Postcondizioni:

- Dati aggiornati.

Flusso di eventi:

1. Il Bibliotecario seleziona il libro.
2. Modifica uno o più campi tramite il form.
3. Conferma cliccando “Modifica libro”.
4. Sistema aggiorna i dati.

Flussi alternativi:

- 2a. Nessun campo modificato → messaggio di notifica
- 2b. Formato dati incorrecto → richiesta di reinserimento
- 3a. Codice ISBN inesistente → messaggio di errore

UC-4 – Cancellazione Libro

Codice: UC-4

Titolo: Cancellazione Libro

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema correttamente avviato.
- Il bibliotecario è autenticato e seleziona “Gestione libri”.
- Almeno un libro esistente nell’inventario.

Flusso di eventi:

1. Il bibliotecario seleziona il libro.
2. Seleziona “Elimina libro”.
3. Sistema rimuove il libro e aggiorna l’inventario.

Flussi alternativi:

- 2a. Tutti i libri in prestito → eliminazione bloccata.

Postcondizioni:

- Libro eliminato.

UC-5 – Ricerca Libro

Codice: UC-5**Titolo:** Ricerca Libro**Attore partecipante:** Bibliotecario**Precondizioni:**

- Sistema correttamente avviato.
- Il bibliotecario è autenticato e seleziona "Gestione libri".

Postcondizioni:

- Risultati mostrati in forma tabellare ordinati per titolo.

Flusso di eventi:

1. Il bibliotecario inserisce i dati su cui filtrare.
2. Seleziona 'Ricerca libro'.
3. Vengono mostrati i risultati ordinati per titolo.

Flussi alternativi:

- 1a. Filtri vuoti → messaggio di notifica.
- 1a. Formato dati filtri incorretti → messaggio di notifica.
- 2a. Nessun libro da mostrare → il sistema mostra una tabella vuota o un messaggio "Nessun libro da mostrare"

2. Gestione Utenti

UC-6 - Visualizzazione elenco utenti

Codice: UC-6

Titolo: Visualizzazione elenco utenti

Attore partecipante: Bibliotecario

Precondizioni:

- Bibliotecario autenticato.
- Sistema correttamente avviato.

Postcondizioni:

- L'elenco utenti è visualizzato in forma tabellare ordinato per cognome e nome.

Flusso di eventi:

1. Il bibliotecario seleziona "Gestione utenti".
2. Il sistema carica dal file/archivio l'elenco degli utenti registrati.
3. Il sistema mostra l'elenco in una tabella con le principali informazioni (matricola, cognome, nome), ordinato per cognome e nome.

Flussi alternativi:

- 2a. Non esistono libri registrati → il sistema mostra una tabella vuota o un messaggio "Nessun libro presente in catalogo".

UC-7 – Inserimento Utente

Codice: UC-7

Titolo: Inserimento Utente

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema avviato correttamente.
- Bibliotecario autenticato e seleziona "Gestione utenti".

Postcondizioni:

- Utente registrato in database.

Flusso di eventi:

1. Il bibliotecario inserisce i dati.
2. Seleziona "Inserisci utente".
3. Il sistema aggiorna database.

Flussi alternativi:

- 1a. Dati mancanti / formato incorretto → richiesta completamento / reinserimento

- 2a. Utente già registrato → inserimento bloccato.

UC-8 – Modifica Utente

Codice: UC-8

Titolo: Modifica Utente

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema avviato correttamente.
- Bibliotecario autenticato e seleziona “Gestione utenti”.
- Almeno un utente registrato in database.

Postcondizioni:

- Dati aggiornati.

Flusso di eventi:

1. Il bibliotecario seleziona l’utente.
2. Modifica campi tramite il form.
3. Seleziona “Modifica libro”.
4. Sistema aggiorna il database.

Flussi alternativi:

- 2a. Nessun campo modificato → messaggio di notifica
- 2b. Formato dati incorretto → richiesta di reinserimento

UC-9 – Cancellazione Utente

Codice: UC-9

Titolo: Cancellazione Utente

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema avviato correttamente.
- Bibliotecario autenticato e seleziona “Gestione utenti”.
- Almeno un utente registrato in database.

Postcondizioni:

- Utente eliminato e database aggiornato.

Flusso di eventi:

1. Il bibliotecario seleziona l’utente da cancellare’.
2. Seleziona “Elimina utente”..
3. Sistema elimina e aggiorna database.

Flussi alternativi:

- 2a. Almeno un prestito attivo → eliminazione bloccata.

UC-10 – Ricerca Utente

Codice: UC-10

Titolo: Ricerca Utente

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema avviato correttamente.
- Bibliotecario autenticato e seleziona "Gestione utenti".

Postcondizioni:

- Risultati mostrati in forma tabellare ordinati per cognome e nome.

Flusso di eventi:

1. Il bibliotecario inserisce i dati su cui filtrare.
2. Seleziona 'Ricerca utente'.
3. Vengono mostrati i risultati ordinati per cognome e nome.

Flussi alternativi:

- 1a. Filtri vuoti → messaggio di notifica.
- 1a. Formato dati filtri incorretti → messaggio di notifica.
- 2a. Nessun utente da mostrare → il sistema mostra una tabella vuota o un messaggio "Nessun utente da mostrare"

3. Gestione Prestiti

UC-11 – Registrazione Prestito

Codice: UC-11

Titolo: Registrazione Prestito

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema avviato correttamente.
- Bibliotecario autenticato e seleziona “Gestione prestiti”.
- Almeno un utente e un libro esistenti.

Postcondizioni:

- Prestito registrato e database e inventario aggiornati.

Flusso di eventi:

1. Il bibliotecario inserisce matricola utente, ISBN libro e data prevista di restituzione.
2. Seleziona “Inserisci prestito”.
3. Sistema registra e aggiorna database e inventario.

Flussi alternativi:

- 1a. Formato dati errato -> notifica di errore.
- 1a. Limite prestiti a disposizione -> notifica di blocco.
- 2a. Libro non disponibile/inesistente -> messaggio di indisponibilità/errore.
- 2b. Utente non esistente -> messaggio di errore

UC-12 – Registrazione Restituzione

Codice: UC-12

Titolo: Registrazione Restituzione

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema avviato correttamente.
- Bibliotecario autenticato e seleziona “Gestione prestiti”.
- Utente con prestito attivo esistente.

Postcondizioni:

- Prestito chiuso e database e inventario aggiornati.

Flusso di eventi:

1. Il bibliotecario seleziona il prestito.
2. Seleziona “Restituisci libro”.
3. Sistema aggiorna database e inventario.

UC-13 – Visualizzazione Elenco Prestiti

Codice: UC-13

Titolo: Visualizzare elenco prestiti

Attore partecipante: Bibliotecario
(autenticato)

Precondizioni:

- Sistema correttamente avviato.
- Bibliotecario autenticato.

Postcondizioni:

- Elenco dei prestiti visualizzato.

Flusso di eventi:

1. Il bibliotecario seleziona "Gestione prestiti".
2. Il sistema recupera l'elenco dei prestiti dal database/file.
3. Il sistema mostra l'elenco dei prestiti in una tabella, ordinato per data di restituzione, evidenziando restituzioni in ritardo.

Flussi alternativi:

- 2a. Non esistono prestiti registrati → il sistema mostra una tabella vuota o un messaggio "Nessun prestito presente".

UC-14 – Gestione Blacklist

Codice: UC-14

Titolo: Gestione blacklist

Attore partecipante: Bibliotecario

Precondizioni:

- Bibliotecario autenticato e seleziona "Gestione prestiti" o "Gestione utente".
- Almeno un utente registrato nel sistema.

Postcondizioni:

- L'utente selezionato risulta contrassegnato come "blacklist: si/no".
- L'utente non può avviare nuovi prestiti finché rimane in blacklist.

Flusso di eventi:

1. Il bibliotecario seleziona l'utente da gestire dall'elenco.
2. Seleziona il comando "Blacklist".
3. Il sistema cambia lo stato blacklist dell'utente.

Flussi alternativi:

- 2a. Nessun utente selezionato → messaggio di errore.

4.Gestione Autenticazione

UC-15 – Login

Codice: UC-15

Titolo: Login

Attore partecipante: Bibliotecario

Precondizioni:

- Sistema correttamente avviato.
- Schermata di login visualizzata.

Postcondizioni:

- Bibliotecario autenticato.
- Schermata iniziale visualizzata.

Flusso di eventi:

1. Il bibliotecario apre l'applicazione e visualizza la schermata di login.
2. Inserisce la password (mascherata).
3. Il sistema verifica le credenziali, autentica il bibliotecario e mostra la schermata iniziale.

Flussi alternativi:

- 2a. Credenziali errate → il sistema mostra un messaggio di errore e richiede un nuovo tentativo.

UC-16 – Logout

Codice: UC-18

Titolo: Effettuare logout

Attore partecipante: Bibliotecario (autenticato)

Precondizioni:

- Bibliotecario autenticato.
- Una schermata dell'applicazione è aperta.

Postcondizioni:

- Sessione del bibliotecario terminata.
- Schermata di login visualizzata.

Flusso di eventi:

1. Il bibliotecario seleziona il comando "Logout".
2. Il sistema chiude la sessione e mostra la schermata di login.

UC-17 – Modifica Password

Codice: UC-19

Titolo: Modifica Password

Attore partecipante: Bibliotecario

Precondizioni:

- La schermata di login dell'applicazione è aperta.

Postcondizioni:

- Schermata di menu visualizzata.
- Password aggiornata.

Flusso di eventi:

1. Il bibliotecario clicca il tasto "Hai dimenticato la password".
2. Viene mostrato un form con una domanda di sicurezza.
3. Il bibliotecario risponde e inserisce la nuova password.
4. Il sistema aggiorna la password e si torna alla schermata di login.

Flusso alternativo:

- 3a. il bibliotecario sbaglia la risposta → il sistema mostra un messaggio di errore e richiede un nuovo tentativo.

5.Persistenza dei Dati

UC-18 – Aggiornamento inventario

Codice: UC-19

Flusso di eventi.

Titolo: Aggiornamento inventario

1. Sistema aggiorna il database dei libri scrivendo su file.

Attore partecipante: Bibliotecario

Precondizioni:

- Operazione di gestione libri

Postcondizioni:

- Inventario aggiornato.

UC-19 – Aggiornamento database utenti

Codice: UC-20

Flusso di eventi.

Titolo: Aggiornamento database utenti

1. Sistema aggiorna il database utenti scrivendo su file.

Attore partecipante: Bibliotecario

Precondizioni:

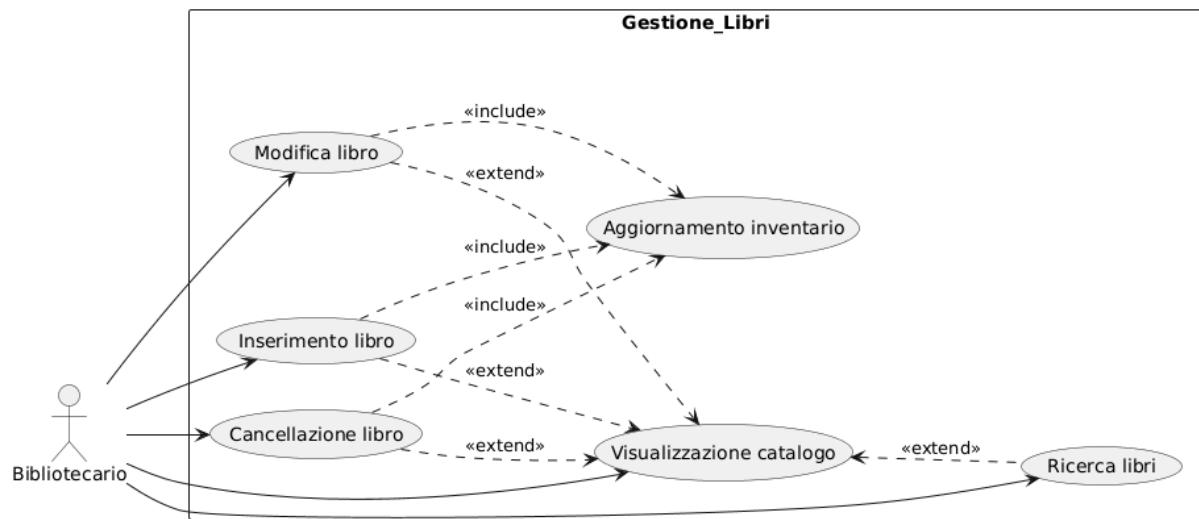
- Operazione di gestione utenti

Postcondizioni:

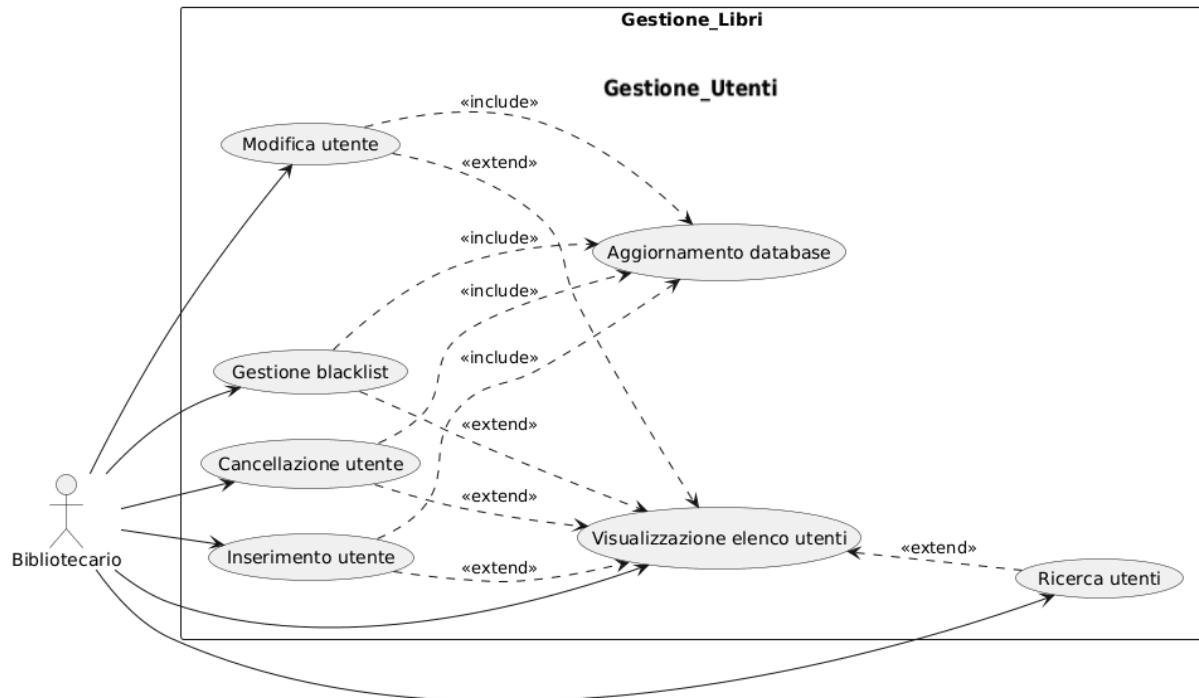
- Database aggiornato.

5. Diagrammi dei Casi d'Uso

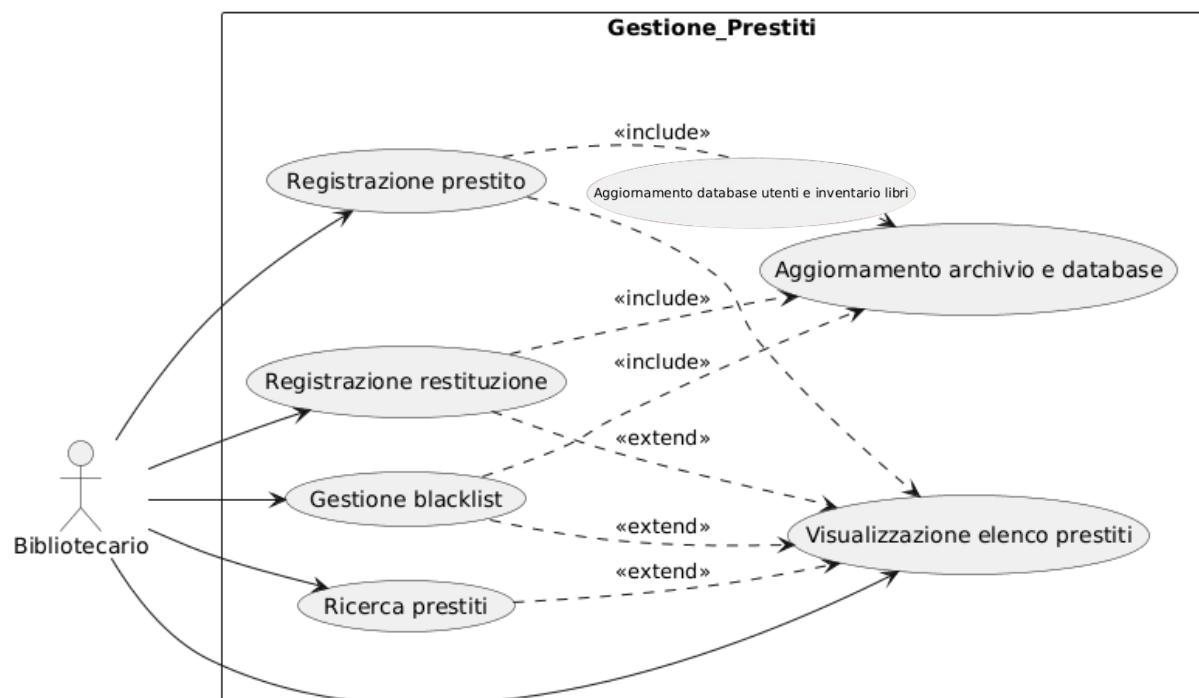
1. Gestione Libri



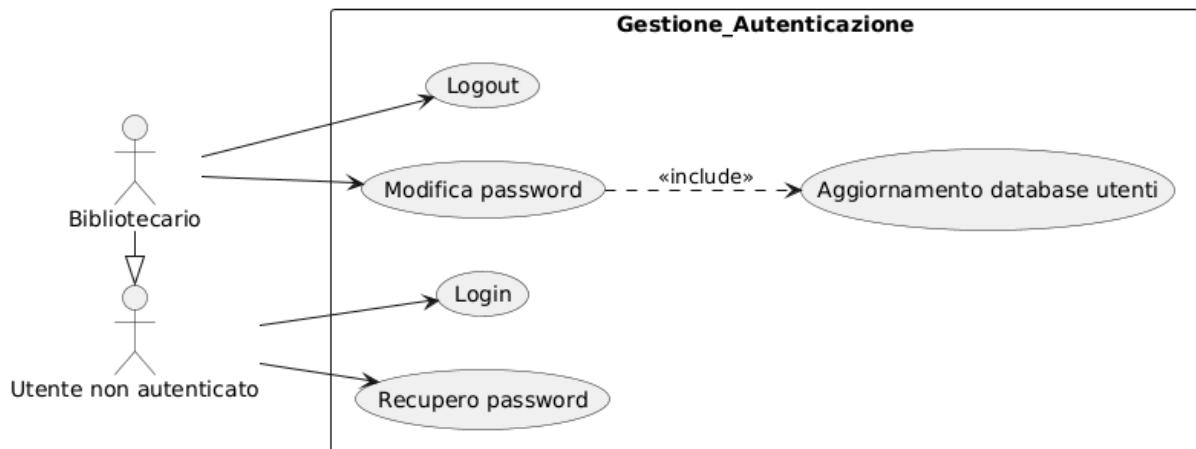
2. Gestione Utenti



3. Gestione Prestiti



4. Gestione Autenticazione



6. Matrice di Tracciabilità

Requisito	Design (package)	Codice (classi/metodi)	Test case	Requisiti correlati
FI-1 Inserire libro	controller, model,view, persistence	LibriController.gestisciInserisci() GestioneLibri.inserisciLibro() ArchivioFile.salvaLibri()	GestioneLibriTest. testInserisciLibroNuovo/testInserisciLibroEsistente_incrementaCopie/testInserisciLibro_copieNonValidelanciaEccezion() ArchivioFileTest. salvaECaricaLibri_ritrovatoTitolo()	DT-1, DT-3, UI-2, VC-5
FI-2 Modificare libro	controller, model,view, persistence	LibriController.gestisciModifica() GestioneLibri.modificaLibro() ArchivioFile.salvaLibri()	GestioneLibriTest. testModificaLibro_modificaCampi/testMo	DT-1, DT-3, UI-2

			dificaLibro_null_nonFaNulla() LibroTest .testSetGetTitolo/testSetGetAnnoPubblicazione/testSetGetAutori/testSetAutoriNull_diventaListaVuota/testSetCopieTotali_aumenta_nonToccaDisponibiliSeGiavalide/testSetCopieTotali_riduce_tagliaDisponibiliSeSuperaNoTotali/testSetCopieDisponibili()	
FI-3 Eliminare libro	controller, model, persistence	LibriController.gestisciElimina(), GestioneLibri.eliminaLibro()	GestioneLibriTest. testEliminaLibro_valido_elimina/testEliminaLibro_stringaVuota_oNull_nonFaNulla/testEliminaLibro_formatoNonValido_noFaNulla()	DT-1,DT-3, VC-4, VC-5
FI-4 Ricercare libro	controller, model,view	LibriController.gestisciCercaOIndietro() GestioneLibri.cercaLibri()	GestioneLibriTest. testCercaLibri_perIsbn/testCercaLibri_isbnNonValido_vieneIgnorato/testCercaLibri_perTitolo_parziale_caseInsensitive/testCercaLibri_perAutore_parziale_caseInsensitive/testCercaLibri_conTuttiFiltri/testCercaLibri_nessunRisultato()	DT-1, UI-2, NFR-3

FI-5 Inserire utente	controller, model, view, persistence	UtentiController.gestisciInserisci(), GestioneUtenti.inserisciUtente(), ArchivioFile.salvaUtenti()	GestioneUtentiTest .testInserisciUtente_ok/testInserisciUtente_null_nonFaNulla() ArchivioFileTest .salvaECaricaUtenti_ritrovNome()	DT-2, DT-3, UI-4, VC-6
FI-6 Modificare utente	view, model, persistence	UtentiController.gestisciModifica(), GestioneUtenti.modificaUtente(), archivioFile.salvaUtenti()	GestioneUtentiTest .testModificaUtente_modificaCampi/testModificaUtente_null_nonFaNulla/testModificaUtente_nonEsiste_nonFaNull() UtenteTest .testSetters()	DT-2, DT-3, UI-4
FI-7 Eliminare utente	view, model, persistence	utentiController.gestisciElimina(), gestionePrestiti.haPrestitiAttivi(), gestioneUtenti.elimnaUtente(), archivioFile.salvaUtenti	GestioneUtentiTest .testEliminaUtente_stringaVuota_oNull_nonFaNulla/testEliminaUtente_nonEsiste_nonFaNulla/testEliminaUtente_senzaPrestiti_attivi_elimina/testEliminaUtente_conPrestitiAttivi_nonElimina()	DT-2, DT-3, VC-4, VC-6
FI-8 Ricercare utente	view, controller, model	UtentiController.gestisciCercaOIndietro() GestioneUtenti.cercaLibri()	GestioneUtentiTest .testCercaUtenti_perMatricola_matchEsatto/testCercaUtenti_perCognome_contains_caseInsensitive/testCercaUtenti_perNome_contains_caseInsensitive/testCercaUten	DT-2, UI-4, NFR-3

			ti_tuttiVuoti_restituisceTutti()	
FI-9 Visualizzare catalogo libri	view,controller,model	MainController.mostraMain, LibriController.inizializzaTabella(), GestioneLibri.getLibriOrdinatiPerTitolo()	GestioneLibriTest. testGetLibri/testGetLibriOrdinatiPerTitolo() LibriControllerTest. inserisciLibro_ok_inserirInModelloETabella,cerca_toggleCercaIndietro,aggiornaDaModel_resetRicercaESelezione MainControllerTest. mostraMain_impostaTitoloEScena	DT-1, UI-2, NFR-1
FI-10 Registrare prestito	view,controller,persistente	PrestitiController.gestisciNuovoPrestito, GestioneUtenti.trovaUtente(), GestioneLibri.trovaLibro(), GestionePrestiti.registraPrestito(), ArchivioFile.salvaPrestiti()	GestionePrestitiTest. testRegistraPrestito_ok_aggiornaTutto/testRegistraPrestito_utenteNull_lancia/testRegistraPrestito_libroNull_lancia/testRegistraPrestito_libroNonDisponibile_lancia/testRegistraPrestito_superaMassimo_lancia() ArchivioFileTest. salvaECaricaPrestiti_listaVuotaRestaVuota() PrestitiControllerTest. nuovoPrestito_ok_registraEaggiornaTabella()	FI-4, FI-8, FI-12, FI-13, DT-2, DT-3, VC-1, VC-2, VC-8, VC-9

FI-11 Registrare restituzione	view, controller,m odel,persiste nce	PrestitiController.gestisciRestit uzione(), GestionePrestiti.registraRestitu zione(), Libro.incrementaCopiaDisponib ile(), ArchivioFile.salvaPrestiti()	PrestitiController Test.restituzione_ok _rimuoveDaAttivi PrestitoTest.testSet DataRestituzione LibroTest.testIncre mentaCopiaDisponibi le_ok/testIncrement aCopiaDisponibile_s eGiaAlMassimo_non SuperaTotali	FI-12, DT-2, DT-3, VC-3, VC-8
FI-12 Controllare disponibilità libri	view, controller, model	PrestitiController.gestisciNuovo Prestito(), GestioneLibri.trovaLibro(), Libro.isDisponibile()	LibroTest.testIsDis ponibile/testDecreme ntaCopiaDisponibile _ok/testDecrementa CopiaDisponibile_qu andoZero_lanciaEcce GestionePrestitiTe st.testRegistraPrestit o_libroNonDisponibi le_lancia/testRegistr aPrestito_ok_aggiorn aTutto	DT-1, VC-2, VC-3, VC-8
FI-13 Controllare limite prestiti utente	view,control ler,model	PrestitiController.gestisciNuovo Prestito(), GestionePrestiti.contaPrestitiAt tivi(), Utente.canNuovoPrestito()	UtenteTest.testCan NuovoPrestitoSucce so/testCanNuovoPres titoBlacklist/testCan NuovoPrestitoMaxRa ggiunto GestionePrestitiTe st.testRegistraPrestit o_superaMassimo_la ncia	DT-2, VC-1, VC-9

FI-14 Visualizzare elenco prestiti	controller,vie w,model	mainController.mostraMain, PrestitiController.inizializzaTab ella(), GestionePrestiti.getPrestitiAttivi()	GestionePrestitiTe st. testGetPrestitiAttivi_inizioVuoto/testC ontaPrestitiAttivi_eHaPrestitiAttiviPer/tes tTrovaPrestitoAttivo _trovato/testTrovaPrestitoAttivo_nonTrov ato PrestitiController Test. nuovoPrestito_ok_registraEaggiorna Tabella() MainControllerTe st. mostraMain_impostaTitoloEScena	DT-2, UI-3, VC-8, NFR-1
FI-15 Inserire utente in blacklist	view,control ler,model	UtentiController.gestisciToggle Blacklist(), PrestitiController.gestisciBlackli st(), GestioneUtenti.setBlackList, ArchivioFile.salvaUtenti()	GestioneUtentiTes t. testSetBlacklist_im postaTrue/testSetBla cklist_utenteNull_no nFaNulla UtenteTest. testSetters	DT-2, UI-3, VC-9
FI-16 Visualizzare elenco utenti	view, controller, model	mainController.mostraMain(), utentiController.inizializzaTabel la(), gestioneUtenti.getUtenti()	GestioneUtentiTes t. testGetUtenti UtentiControllerT est. aggiornaDaModel _resetRicercaESelezione() MainControllerTe st. mostraMain_impostaTitoloEScena()	DT-2, UI-4, NFR-1
FI- 17 Login	view, controller,	ArchivioFile.caricaAutenticazio ne(), AuthController.login() Autenticazione.login(), MainController.mostraMenu(),	AuthControllerTes t. login_ok_vaAlMen u_pulisceCampi_eAz zeraErrore/login_err	FI-18, FI-21, UI-1,

	model, presistence	LoginView.mostraErrore(), LoginView.pulisciCampi()	or_passwordErrata_ mostraErrore_ePulis ceCampi ArchivioFileTest.s alvaECaricaAutentica zione_passwordFunz iona() MainControllerTe st.mostraLogin_im ostaTitoloEScena()	NFR-6, NFR-7
FI-18 Mascheramento Password	view	LoginView.campoPasswordVisibile.textProperty().bindBidirectional(), LoginView.checkMostraPassword.selectedProperty().addListener(...), campoPasswordVisibile.setVisible(...)	LoginViewTest.tog gleMostraPassword_ mostraTextFieldENas condePasswordField/ bindingBidirezionale _passwordSempreSin cronizzata	FI-17, FI-19, NFR-1, NFR-7
FI-19 Modifica Password	view	LoginView.getPassword()	LoginViewTest.pas swordFieldPresente_ eUsatoPerLaPasswor d	FI-17 , NFR-7
FI- 21 Logout	view, controller	MainController.mostraLogin()	MainControllerTe st.mostraLogin_im ostaTitoloEScena()	FI-17, UI-1, NFR-2, NFR-7
DT-1 Gestire dati libro	model, persistence	GestioneLibri.getLibri(), trovaLibro(), cercaLibri(), ArchivioFile.caricaLibri(), salvaLibri()	tutte i metodi di GestioneLibriTest/LibroTest ArchivioFileTest.c aricaLibri_senzaFile _restituisceVuoto/sal vaECaricaLibri ritro voTitolo/caricaLibri fileCorrotto_restituis ceVuoto/salvaLibri_n ull_nonCreaFile/salv aLibri_seFileEUnaCa rtella_nonLancia	FI-1..FI-4 , FI-9, VC-5

DT-2 Gestire dati utente	mode, persistence	GestioneUtenti.getUtenti, trovaUtente, cercaUtenti(), setBlackList() ArchivioFile.caricaUtenti(), salvaUtenti()	Tutte i metodi di GestioneUtenteTest e UtenteTest ArchivioFileTest.c aricaUtenti_senzaFil e_restituisceVuoto/s alvaECaricaUtenti_r i trovoNome	FI-5..FI-8 , FI-10, FI-11, FI-14..FI- 16, VC-6, VC-8
DT-3 Archivio persistente	model, persistence	ArchivioFile.caricaAutenticazio ne(), ArchivioFile.salvaAutenticazion e(), Autenticazione.getPasswordHas h(), Autenticazione.calcolaHash()	ArchivioFileTest.s alvaECaricaAutentica zione_passwordFunz iona	FI-1..FI-1 6, UI-1, NFR-2, NFR-6, VC-7
UI-1 Schermata iniziale/men u	controller,vie w,	MainController.mostraMenu()	AuthControllerTes t.login_ok_vaAlMen u_pulisceCampi_eAz zeroErrore() MainControllerTe st.mostraMenu_im postaTitoloEScena()	FI-1..FI-1 6, NFR-1
UI-2 Schermata gestione libri	view,controll er,model	LibriController.collegaeventi(lib riController.gestisciInserisci/m odifica/elimina/cercaOIndietro(), gestioneLibri(), LibriPanel.getLibroSelezionato(), LibriPanel.setCampiDaRiga()	LibriControllerTe st.aggiornaDaModel _resetRicercaESelezi one()	FI-1..FI-4 , FI-9, DT-1, NFR-1, NFR-3
UI-3 Schermata gestione prestiti	view,controll er,model	PrestitiPanel.(mostraPrestiti(), getPrestitoSelezionato(), mostraMessaggio()), PrestitiController.(gestisciNuov oPrestito(), gestisciRestituzione(), gestisciBlacklist())	PrestitiController Test.nuovoPrestito_ ok_registraEaggiorna Tabella()	FI-10...FI -15, DT-2, VC-1..VC- 3, VC-9

UI-4 Schermata gestione utenti	view, controller,model	UtentiPanel.(get...campi(), getUtenteSelezionato(), setCampiDaRiga()) UtentiController.(gestisciInserisci()/Modifica()/Elimina()/ToggleBlacklist()/CercaOIndietro)()	UtentiControllerTest. aggiornaDaModel _resetRicercaESelezione()	FI-5..FI-8 , FI-16, DT-2, NFR-1
NFR-1 Usabilità interfaccia	view	tutte le classi della view	è possibile testarlo facendo test manuali.	UI-1..UI-4, FI-1..FI-16
NFR-2 Affidabilità dati	persistence, model	archivioFile.carica(), archivioFile.salva(), tutte le classi del model	UtenteTest. testAggiungiPrestitoDuplicato() GestionePrestitiTest. testRegistraRestituzione_suPrestitoGiaRestituito_nonDoppiaIncremento() LibroTest. testAutori_copiaDifensiva_costruttore/testAutori_copiaDifensiva_setter/testSetCopieTotali_reduce_tagliaDisponibiliSeSuperanoTotali() PrestitoTest. testIsInRitardoFalsoSeNonScaduto/testIsInRitardoFalsoSeGiornoEsatto/testIsInRitardoVerSeScaduto/testIsInRitardoFalsoSeRestituito() ArchivioFileTest. salvaECaricaLibri_ritrovatoTitolo/salvaECari	DT-1..DT-3, VC-7

			caUtenti ritrovoNome/salvaECaricaPrestiti_listVuotaRestaVuota/salvaECaricaAutenticazione_passwordFunziona()	
NFR-3 Prestazioni ricerca	model, controller	GestioneLibri utilizza Treeset con cercaLibri, compareTO e la stessa cosa anche GestioneUtenti	<p>GestioneLibriTest. testCercaLibri_perIsbn/testCercaLibri_isbnNonValido_vieneIgnorato/testCercaLibri_perTitolo_parziale_caseInsensitive/testCercaLibri_perAutorе_parziale_caseInsensitive/testCercaLibri_conTuttiFiltri/testCercaLibri_nessunRisultato()</p> <p>GestioneUtentiTest. testCercaUtenti_perMatricola_matchEsatto/testCercaUtenti_perCognome_contains_caseInsensitive/testCercaUtenti_perNome_contains_caseInsensitive/testCercaUtenti_tuttiVuoti_restituisceTutti()</p>	FI-4, FI-8, FI-9, FI-14, UI-2..UI-4
NFR-4 Manutenibilità	view,model,controller,main,persistence	tutte le classi	<p>GestioneUtentiTest. testSetGestionePrestiti()</p> <p>MainControllerTest. setPrestitiController_impostaCampo()</p>	tutti i moduli

NFR-5 Portabilità	persistence	ArchivioFile	ArchivioFileTest.c aricaLibri_senzaFile _restituisceVuoto/ca ricaUtenti_senzaFile _restituisceVuoto/ca ricaPrestiti_senzaFile _restituisceVuoto()	DT-3, tutti i moduli
NFR-6 Robustezza/g estione errori	controller,m odel	LibriController, UtentiController, PrestitiController	GestionePrestitiTe st.testRegistraPrestit o_utenteNull_lancia/ testRegistraPrestito_1 ibroNull_lancia/test RegistraPrestito_libr oNonDisponibile_lan cia/testRegistraPresti to_superaMassimo_l ancia/testRegistraRe stituzione_null_nonF aNulla() GestioneLibriTest. testCostruttoreConTr eeSetNull/testInseris ciLibro_copieNonVal ide_lanciaEccezione/ testModificaLibro_n ull_nonFaNulla/test EliminaLibro_stringa Vuota_oNull_nonFa Nulla/testEliminaLib ro_formatoNonValid o_nonFaNulla/testCe recaLibri_isbnNonVal ido_vieneIgnorato() GestioneUtentiTes t.testCostruttoreCon TreeSetNull/testTrov aUtente_null/testIns erisciUtente_null_no nFaNulla/testModific aUtente_null_nonFa	FI-1..FI-8 , DT-1..DT- 3

			<p>Nulla/testEliminaUte nte_stringaVuota_o Null_nonFaNulla/tes tEliminaUtente_non Esiste_nonFaNulla/t estSetBlacklist_utent eNull_nonFaNulla()</p> <p>LibroTest.testSetAu toriNull_diventaLista Vuota/testDecrement aCopiaDisponibile_q uandoZero_lanciaEcc ezione()</p> <p>UtenteTest.testAggi ungiPrestitoNull/test CompareToNull()</p> <p>PrestitoTest.testIsI nRitardoInputNull()</p> <p>AuthControllerTes t.loginErrato_mostra ErroreEPulisce() ArchivioFileTest.c aricaLibri_fileCorrott o_restituisceVuoto/s alvaLibri_null_nonC reaFile/salvaLibri_se FileEUnaCartella_no nLanci tuti test model</p>	
NFR-7 Sicurezza/int egrità dati	model, persistence e vincoli	Autenticazione.calcolaHash(), login();	<p>AuthControllerTes t.login_ok_pulisceCa mpi_e_vaAlMenu,Au thControllerTest.logi n_ko_mostraErrore_ e_pulisceCampi</p> <p>LibroTest.testGetAu toriUnmodifiable()</p>	DT-1..DT- 3, VC-7, VC-8

			UtenteTest .testGetPrestitiAttiviUnmodifiable() ArchivioFileTest .salvaECaricaAutenticazione_passwordFunziona()	
VC-1 Max 3 prestiti per utente	controller, model	PrestitiController.gestisciNuovoPrestito(), GestionePrestiti.contaPrestitiAttivi(), Utente.canNuovoPrestito()	UtenteTest .testCanNuovoPrestitoSuccesso/testCanNuovoPrestitoMaxRaggiunto() GestionePrestitiTest .testRegistraPrestito_superaMassimo_lancia() PrestitiControllerTest .aggiornaDaModel_riempieLaTabellaConDuePrestiti	FI-10, FI-13, UI-3
VC-2 Copie disponibili per prestito	view, controller,model	prestitiController.gestisciNuovoPrestito(), gestioneLibri.trovaLibro(), libro.isDisponibile.	LibroTest .testIsDisponibile/testDecrementaCopiaDisponibile_ok/testDecrementaCopiaDisponibile_quandoZero_lanciaEccezione() GestionePrestitiTest .testRegistraPrestito_ok_aggiornaTutto/testRegistraPrestito_1ibroNonDisponibile_lancia()	FI-10, FI-12
VC-3 Aggiornamento copie alla restituzione	view, controller,model	prestitiController.gestisciRestituzione(), gestionePrestiti.registraRestituzione(),	GestionePrestitiTest .testRegistraRestituzione_ok_rendePrestitoNonAttivo/testRegistraRestituzione_suPrestitoGiàRestituito	FI-11, FI-12

		libro.incrementaCopiaDisponibile	_nonDoppiaIncremento() LibroTest .testIncrementaCopiaDisponibile_ok/testIncrementaCopiaDisponibile_seGiaAlMassimo_nonSuperaTotal()	
VC-4 Divieto eliminazione con prestiti attivi	controller,model	UtentiController.gestisciElimina(), GestionePrestiti.haPrestitiAttiviPer(utente), libriController.gestisciElimina()	LibriControllerTest .eliminaLibro_errore senzaSelezione_nonElimina, UtentiControllerTest .elimina_error_se nzaMatricolaENessunaSelezione_nonEli mina	FI-3, FI-7, FI-14
VC-5 ID libro univoco	model	libro.compareTo,Libro.equals/hashCode(), GestioneLibri.trovaLibro.	LibroTest .testCompareTo/testEquals/testHashCode() GestioneLibriTest .testInserisciLibroEsiste_incrementeCopie()	DT-1, FI-1, FI-2
VC-6 ID utente univoco	model	Utente.compareTo,Utente.equals/hashCode(), GestioneUtente.trovaLUtente.	UtenteTest .testEquals/testHashCode/testCompareTo/testCompareToNull()	DT-2, FI-5, FI-6
VC-7 Integrità archivio	persistence, model	archivioFile.salvaLibri(),archivioFile.salvaUtenti(), archivioFile.salvaPrestiti(), archivioFile.salvaAutenticazione e carica e salva.	ArchivioFileTest .salvaECaricaLibri ritrovatoTitolo/salvaECaricaUtenti ritrovoNome/salvaECaricaPrestiti listaVuotaRestaVuita/salvaECaricaAutenticazione_passwordF	DT-1..DT-3, NFR-2

			unziona/caricaLibri_fileCorrotto_restituisceVuoto/salvaLibri_null_nonCreaFile/salvaLibri_seFileEUnaCartella_nonLancia	
VC-8 Prestiti solo per soggetti registrati	view,controller,model	PrestitiControllerTest.gestisciNuovoPrestito_utenteNonRegistrato_nonRegistraPrestito PrestitiControllerTest.gestisciNuovoPrestito_libroNonPresente_nonRegistraPrestito	PrestitiControllerTest.nuovoPrestito_error_utenteNonRegistrato_nonRegistraPrestito	FI-10, FI-11, DT-1, DT-2
VC-9 Rispetto regolamento biblioteca	controller,model	Utente.IsInBlacklist(),prestitiController.gestisciNuovoPrestito(), Utente.canNuovoPrestito()	UtenteTest.testSetters/testCanNuovoPrestitoBlacklist() GestioneUtentiTest.testSetBlacklist_impostaTrue()	VC-1..VC-4, FI-10..FI-15