

Giorgia Rispoli [0612709246]

Giorgio Lupo [0612709260]

Margherita Gerarda Morriello [0612704256]

Emanuele Troisi [0612709575]

DOCUMENTO DEI REQUISITI

Biblioteca universitaria - GRUPPO 25

INDICE

DESCRIZIONE DELL'APPLICAZIONE.....	1
REQUISITI FUNZIONALI E NON FUNZIONALI:DESCRIZIONE GENERALE	1.1
CLASSIFICAZIONE DEI REQUISITI.....	1.2
GESTIONE DEI REQUISITI.....	2
REQUISITI FUNZIONALI.....	2.1
INTERFACCIA GRAFICA.....	2.2
REQUISITI NON FUNZIONALI.....	2.3
DEFINIZIONE DELLE PRIORITA'.....	2.4
TABELLA DELLE PRIORITA'.....	2.5
CASI D'USO.....	3
DESCRIZIONE DEI CASI D'USO.....	3.1
DIAGRAMMA DEI CASI D'USO.....	3.2
MATRICE DI TRACCIABILITA'.....	4

1 Descrizione dell'Applicazione

Il documento in seguito si avvale di descrivere i requisiti funzionali e non funzionali per l'organizzazione di una biblioteca universitaria con gli opportuni casi d'uso che compongono il sistema. Tale gestione di biblioteca consente la registrazione di libri ed utenti con opportuni dati specifici: l'inserimento di un libro avviene mediante compilazione di un form, aggiungendo elementi come *titolo del libro, nome dell'autore, codice ISBN (o ulteriori identificativi), anno di pubblicazione e numero di copie disponibili al prestito*. Analogamente alla procedura di inserimento di un nuovo libro, è possibile inserire uno studente compilando i campi di *nome, cognome, matricola, e-mail istituzionale e lista dei libri attualmente in prestito*, quest'ultima con data di restituzione. La praticità e la rapidità d'uso del sistema da parte dell'utente è garantita grazie ad un'interfaccia grafica, mediante la quale il bibliotecario (attore coinvolto), dopo aver effettuato l'accesso, può visualizzare nella schermata principale liste ordinate di libri ed utenti e le rispettive azioni che possono essere svolte come prestiti e restituzioni con gli opportuni aggiornamenti. Nello specifico, l'utente può individuare il libro desiderato svolgendo per esso azioni di modifica di dati bibliografici quando necessario, di eliminazione se voluta, e di inserimento nel momento in cui viene pubblicata una nuova opera. Per la gestione degli studenti che possono disporre dei volumi presenti in biblioteca, il sistema offre l'opportunità di modificare i dati anagrafici e le ulteriori informazioni personali di ognuno in caso di variazioni, di eliminazione dall'archivio in caso di necessità o di ricerca tramite nome, cognome e matricola. Il sistema risulta essere capace di gestire eventuali incoerenze di dati, facendo rispettare vincoli di validità (ad esempio conformità o assenza di caratteri speciali nei campi nome) in fase di inserimento. Una delle caratteristiche più significative del sistema è quella di gestire i prestiti e le restituzioni: lo studente che in biblioteca richiede la copia di un libro viene segnalato ed i suoi dati vengono aggiornati. In particolare allo studente vengono aggiunte tra i dati la lista dei libri attualmente in prestito con la relativa data di restituzione. Il sistema può segnalare eventuali anomalie, riguardanti sia i libri che gli studenti tra cui la volontà di richiedere in prestito un quarto libro (ogni studente può avere al massimo tre libri in prestito contemporaneamente) e il prestito di un libro che non è disponibile in biblioteca causa esaurimento di copie. L'applicazione deve essere oltre che di facile utilizzo, capace di gestire un ingente quantità di dati che possono essere le molteplici copie di uno stesso libro (il cui numero viene decrementato quando viene prestato allo studente) o il numero di studenti che possono appartenere ad un'università.

L'intero archivio della biblioteca (libri, utenti e prestiti) è salvato su un file.

1.1 Requisiti funzionali e non funzionali: descrizione generale

Mediante i requisiti fondamentali, possono essere elicitate le informazioni fondamentali per la corretta realizzazione del software facilitandone la comprensione e l'ordine. Le specifiche dei requisiti possono essere suddivise in due fondamentali categorie:

- **Requisiti funzionali:** descrivono le funzioni che il sistema deve essere in grado di eseguire (quindi cosa deve fare il programma)
- **Requisiti non funzionali:** vincoli che descrivono come deve operare il programma

1.2 Classificazione dei requisiti

In seguito, le categorie secondo le quali sono organizzati i requisiti:

BF: Business Flow

IF: Funzionalità Individuali

DF: Dati e Formato dei dati

UI: Interfaccia grafica

FC: Vincoli di qualità

2 Gestione dei requisiti

2.1 Requisiti funzionali

BF-1: Gestione dei libri

- IF-1.1:** l'utente può inserire un nuovo libro specificando tutti i campi richiesti dal modulo
- IF-1.2:** l'utente può modificare i dati di un libro esistente
- IF-1.3:** l'utente può cancellare un libro dall'archivio
- IF-1.4:** l'utente può cercare un libro per:
 - Titolo
 - Autore
 - Codice identificativo
- DF-1.1:** ogni libro deve contenere obbligatoriamente:
 - titolo;
 - autore/i (nome e cognome);
 - anno di pubblicazione;
 - codice di identificazione (ISBN o simili);
 - numero di copie disponibili
- DF-1.2:** il codice identificativo deve essere univoco
- DF-1.3:** il numero di copie non può essere negativo
- DF-1.4:** non sono accettati caratteri speciali per i seguenti campi:

- autore/autori
- codice di identificazione
- numero di copie disponibili
- i. **DF-1.5:** non sono accettati valori identificativi di date non ancora giunte per i seguenti campi:
- anno di pubblicazione

BF-2: Gestione studenti

- a. **IF-2.1:** L'utente può inserire un nuovo studente con i relativi dati personali
- b. **IF-2.2:** L'utente può modificare i dati di uno studente registrato
- c. **IF-2.3:** L'utente può eliminare uno studente dall'archivio
- d. **IF-2.4:** L'utente può cercare uno studente per cognome o matricola
- e. **DF-2.1:** Ogni studente deve includere obbligatoriamente nell'anagrafica:
 - i. nome;
 - ii. cognome;
 - iii. matricola;
 - iv. e-mail istituzionale;
 - v. lista dei libri attualmente in prestito;
 - vi. data di restituzione del libro;
- f. **DF-2.2:** La matricola deve essere univoca;
- g. **DF-2.3:** non sono accettati caratteri speciali per i seguenti campi:
 - nome
 - cognome
 - matricola

BF-3: Gestione dei prestiti

- a. **IF-3.1:** Il sistema deve consentire la registrazione di un nuovo prestito specificando i campi del modulo
- b. **IF-3.2:** Deve essere consentito il prestito solo se vi sono copie disponibili del libro richiesto
- h. **IF-3.3:** Ogni studente può avere al massimo tre prestiti attivi;
- i. **IF-3.4:** Il sistema deve decrementare il numero di copie disponibili ogni volta che viene effettuato un prestito
- c. **IF-3.5:** Il sistema deve visualizzare l'elenco dei prestiti attivi, ordinato per data prevista di restituzione (evidenziando eventuali ritardi)
- d. **IF-3.6:** Il sistema deve incrementare il numero di copie ogni volta che viene segnalata una restituzione

2.2 Requisiti Interfaccia Grafica

UI-1: Interfaccia gestione libri

L'interfaccia deve fornire un form completo per inserire un nuovo libro compilando i dati obbligatori e segnalando eventuali errori di compilazione.

L'utente deve poter selezionare un libro dall'elenco e modificarne i dati attraverso una schermata dedicata.

L'interfaccia deve permettere la cancellazione di un libro.

L'interfaccia deve mostrare l'elenco dei libri ordinati per titolo, aggiornato in seguito a operazioni di modifica.

L'interfaccia deve consentire di cercare un libro tramite:

- titolo,
- autore,
- codice identificativo.

UI-2: Interfaccia gestione studenti

L'interfaccia deve fornire un form completo per inserire un nuovo studente compilando i dati obbligatori e segnalando eventuali errori di compilazione.

L'utente deve poter selezionare uno studente dall'elenco e modificarne i dati attraverso una schermata dedicata.

L'interfaccia deve permettere la rimozione di uno studente.

L'interfaccia deve visualizzare l'elenco completo degli studenti registrati, ordinato per cognome.

L'interfaccia deve consentire la ricerca per cognome o matricola.

UI-3: Interfaccia gestione prestiti

L'utente deve poter selezionare uno studente e un libro, indicando la data prevista di restituzione.

L'interfaccia deve impedire o segnalare:

- prestito di un libro senza copie disponibili,
- prestito a studenti che hanno già 3 prestiti attivi.

L'interfaccia deve mostrare l'elenco dei prestiti ordinati per data prevista di restituzione.

L'interfaccia deve permettere di registrare la restituzione di un libro, aggiornando:

- lo stato del prestito,
- il numero di copie disponibili.

2.3 Requisiti non funzionali

FC-1 Manutenibilità: il sistema deve essere ben organizzato, con un codice chiaro ed intuitivo in modo da renderlo facilmente modificabile ed ordinato favorendone la manutenzione

FC-2 Prestazioni: il sistema deve essere in grado di supportare e gestire automaticamente grandi quantità di dati (quelli che mediamente possono riguardare un'università) fornendo in modo esplicito aggiornamenti in tempo reale sulle azioni che coinvolgono i libri. Il caricamento della pagina iniziale deve completarsi entro i 10 secondi

FC-3 Usabilità: il sistema deve disporre di un'interfaccia facilmente intuibile e semplice da utilizzare, capace di fornire, con l'opportuno design, una visualizzazione corretta dei risultati. Il sistema deve poi fornire messaggi di errore efficaci in caso di input non valido

FC-4 Compatibilità: il sistema deve essere eseguibile su un sistema operativo capace di supportare le ultime versioni di Java

2.4 Definizione delle priorità

Di seguito la definizione delle priorità di ciascun requisito, utile ad ordinare per importanza seguendo il criterio del Business value:

```
-----
--
BUSINESS VALUE - INDICE DI PRIORITA'
MUST HAVE (priorità alta)
SHOULD HAVE (priorità media)
NICE TO HAVE (priorità bassa)
-----
--
RISCHIO TECNICO - INDICE DI PRIORITA'
ALTO
MEDIO
BASSO
```

2.5 Tabella delle priorità

REQUISITO	FUNZIONALE/ NON FUNZIONALE	RISCHIO TECNICO	PRIORITA'
IF-1.1	FUNZIONALE	BASSO	MUST HAVE
IF-1.2	FUNZIONALE	BASSO	MUST HAVE
IF-1.3	FUNZIONALE	MEDIO	MUST HAVE
IF-1.4	FUNZIONALE	BASSO	MUST HAVE
IF-2.1	FUNZIONALE	BASSO	MUST HAVE
IF-2.2	FUNZIONALE	BASSO	SHOULD HAVE
IF-2.3	FUNZIONALE	MEDIO	MUST HAVE
IF-2.4	FUNZIONALE	BASSO	SHOULD HAVE
IF-3.1	FUNZIONALE	BASSO	MUST HAVE
IF-3.2	FUNZIONALE	BASSO	MUST HAVE
IF-3.3	FUNZIONALE	MEDIO	SHOULD HAVE
IF-3.4	FUNZIONALE	MEDIO	MUST HAVE
IF-3.5	FUNZIONALE	BASSO	SHOULD HAVE
IF-3.6	FUNZIONALE	BASSO	MUST HAVE
DF-1.1	FUNZIONALE	BASSO	MUST HAVE
DF-1.2	FUNZIONALE	BASSO	MUST HAVE
DF-1.3	FUNZIONALE	BASSO	SHOULD HAVE
DF-1.4	FUNZIONALE	BASSO	NICE TO HAVE
DF-1.5	FUNZIONALE	BASSO	MUST HAVE
DF-2.1	FUNZIONALE	BASSO	MUST HAVE
DF-2.2	FUNZIONALE	BASSO	SHOULD HAVE
DF-2.3	FUNZIONALE	BASSO	NICE TO HAVE
FC-1	NON FUNZIONALE	BASSO	MUST HAVE
FC-2	NON FUNZIONALE	BASSO	MUST HAVE

FC-3	NON FUNZIONALE	BASSO	SHOULD HAVE
FC-4	NON FUNZIONALE	BASSO	MUST HAVE
UI-1.1	FUNZIONALE	BASSO	MUST HAVE
UI-1.2	FUNZIONALE	BASSO	MUST HAVE
UI-1.3	FUNZIONALE	BASSO	MUST HAVE

3 Casi d'uso

3.1 Descrizione casi d'uso

UC-1	Gestione libri
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • archivio libri caricato.
Postcondizioni	<ul style="list-style-type: none"> • archivio libri aggiornato in base alle operazioni eseguite.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario apre "Gestione libri". 2. Il sistema mostra l'elenco dei libri ordinato per titolo. 3. Il bibliotecario sceglie una delle operazioni: Cercare libro, Inserire libro, Modificare libro, Eliminare libro. 4. Il sistema esegue il caso d'uso selezionato e aggiorna l'archivio.

UC-2	Cercare libro
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • archivio libri caricato.
Postcondizioni	<ul style="list-style-type: none"> • elenco filtrato secondo i criteri di ricerca.

Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona "Cercare libro". 2. Inserisce uno o più criteri (titolo, autore, codice identificativo). 3. Il sistema mostra i libri che soddisfano i criteri.
--------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

UC-3	Inserire libro
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • form di inserimento aperto.
Postcondizioni	<ul style="list-style-type: none"> • nuovo libro aggiunto all'archivio, se i dati sono validi.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona "Inserire libro". 2. Compila i campi obbligatori (titolo, autore/i, anno, codice identificativo univoco, numero copie ≥ 0). 3. Il sistema verifica i vincoli (univocità codice, caratteri ammessi, anno non futuro). 4. In caso di esito positivo, il sistema registra il libro e aggiorna l'elenco.
Flusso alternativo	<ol style="list-style-type: none"> 1. se i dati non rispettano i vincoli, viene attivato "Errore: input non valido".

UC-4	Modificare libro
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • esiste almeno un libro in archivio.
Postcondizioni	<ul style="list-style-type: none"> • i dati del libro selezionato risultano aggiornati.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona "Modificare libro" e sceglie un libro dall'elenco. 2. Modifica uno o più campi rispettando i vincoli sui dati. 3. Il sistema salva le modifiche e aggiorna l'elenco.
Flusso alternativo	<ol style="list-style-type: none"> 1. dati non validi: "Errore: input non valido".

UC-5	Eliminare libro
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • esiste il libro da eliminare.
Postcondizioni	<ul style="list-style-type: none"> • il libro non compare più nell'archivio.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona "Eliminare libro" e sceglie un libro. 2. Conferma l'eliminazione. 3. Il sistema rimuove il libro e aggiorna l'elenco.

UC-6	Errore: input non valido
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • attivato da Inserire/Modificare libro o dagli analoghi sui studenti.
Postcondizioni	<ul style="list-style-type: none"> • nessuna modifica applicata ai dati; • l'errore è stato comunicato.
Flusso principale	<ol style="list-style-type: none"> 1. Il sistema rileva un input che viola i vincoli. 2. Mostra un messaggio di errore e richiede la correzione. 3. Il bibliotecario può ripetere l'operazione con dati corretti.

UC-7	Gestione utenti
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • archivio studenti caricato.
Postcondizioni	<ul style="list-style-type: none"> • archivio studenti aggiornato.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario apre "Gestione utenti". 2. Il sistema mostra l'elenco studenti (ordinata per nome e cognome) 3. Seleziona una delle operazioni: Cercare studente, Inserire studente, Modificare studente, Eliminare studente. 4. Il sistema esegue il caso selezionato e aggiorna l'elenco.

UC-8	Cercare studente
-------------	-------------------------

Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • archivio studenti caricato.
Postcondizioni	<ul style="list-style-type: none"> • elenco filtrato.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario inserisce cognome o matricola. 2. Il sistema mostra gli studenti corrispondenti

UC-9	Inserire studente
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • form di inserimento aperto.
Postcondizioni	<ul style="list-style-type: none"> • nuovo studente aggiunto all'archivio.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona "Inserire studente". 2. Compila i campi obbligatori (nome, cognome, matricola univoca, e-mail istituzionale). 3. Il sistema verifica i vincoli (assenza caratteri speciali, matricola univoca). 4. In caso positivo registra lo studente.

UC-10	Modificare studente
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • lo studente esiste in archivio.
Postcondizioni	<ul style="list-style-type: none"> • dati anagrafici aggiornati.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona "Modificare studente" e sceglie uno studente. 2. Aggiorna i dati. 3. Il sistema controlla i vincoli e salva le modifiche.

UC-11	Eliminare studente
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • studente presente in archivio.
Postcondizioni	<ul style="list-style-type: none"> • lo studente è rimosso dall'archivio.

Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona “Eliminare studente”. 2. Conferma l’eliminazione.
--------------------------	--------------------------------------------------------------------------------------------------------------------------------------------

UC-12	Gestione prestiti
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • archivi libri e studenti caricati.
Postcondizioni	<ul style="list-style-type: none"> • elenco prestiti e numero copie aggiornati.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario apre “Gestione prestiti”. 2. Seleziona uno studente e un libro oppure richiede la sola visualizzazione dei prestiti. 3. Sceglie se visualizzare prestiti, registrare un nuovo prestito o registrare una restituzione. 4. Il sistema esegue il caso incluso corrispondente e aggiorna dati di prestiti e copie.

UC-13	Visualizzare prestiti
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • esistono prestiti registrati.
Postcondizioni	<ul style="list-style-type: none"> • nessuna modifica ai dati; solo visualizzazione.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona “Visualizzare prestiti”. 2. Il sistema mostra l’elenco dei prestiti attivi ordinato per data prevista di restituzione, evidenziando eventuali ritardi.

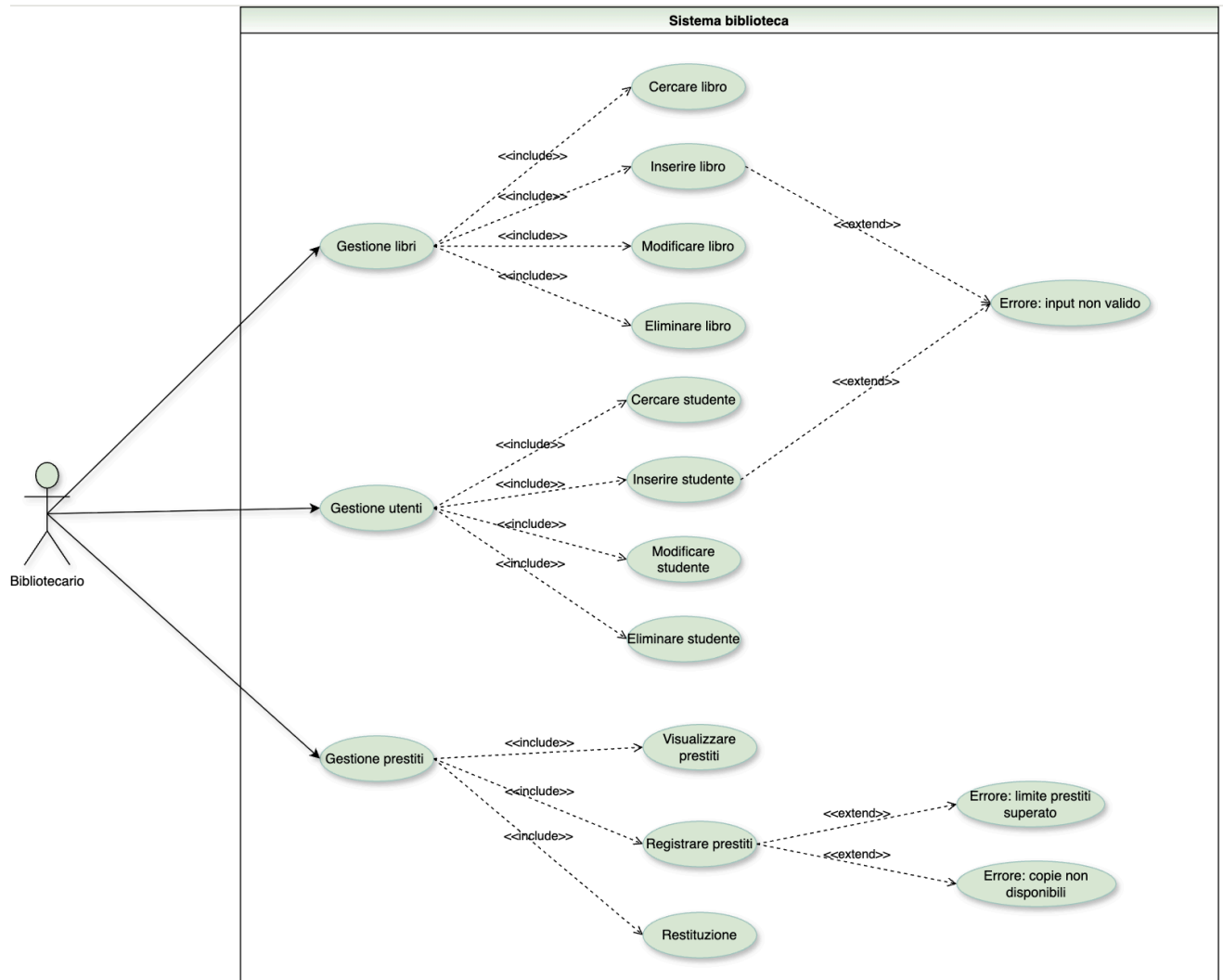
UC-14	Registrare prestito
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • studente valido selezionato; • libro esistente.
Postcondizioni	<ul style="list-style-type: none"> • nuovo prestito registrato; • copie libro decrementate.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona “Registrare prestiti”. 2. Seleziona studente e libro e indica la data prevista di restituzione. 3. Il sistema controlla che lo studente abbia meno di tre prestiti attivi e che esistano copie disponibili.

	4. Se i controlli vanno a buon fine, il sistema registra il prestito, aggiorna la lista prestiti dello studente e decrementa il numero di copie.
Flusso alternativo	<ol style="list-style-type: none"> 1. Se lo studente ha già tre prestiti attivi → attiva “Errore: limite prestiti superato”. 2. Se non ci sono copie disponibili → attiva “Errore: copie non disponibili”.

UC-15	Restituzione
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • esiste un prestito attivo per quello studente e quel libro.
Postcondizioni	<ul style="list-style-type: none"> • prestito chiuso; • numero copie incrementato; • eventuale multa segnalata.
Flusso principale	<ol style="list-style-type: none"> 1. Il bibliotecario seleziona “Restituzione” per uno specifico prestito. 2. Il sistema aggiorna lo stato del prestito a restituito e incrementa il numero di copie disponibili del libro. 3. Se la restituzione avviene oltre le due settimane previste, il sistema segnala una multa associata allo studente.

UC-16	Errore: limite prestiti superato
Attore	Bibliotecario
Precondizioni	<ul style="list-style-type: none"> • chiamato da “Registrare prestiti”
Postcondizioni	<ul style="list-style-type: none"> • nessun nuovo prestito registrato.
Flusso principale	<ol style="list-style-type: none"> 1. Il sistema rileva il superamento del limite. 2. Mostra un messaggio di errore e annulla l'operazione.

3.2 Diagramma dei casi d'uso



4 MATRICE DI TRACCIABILITÀ

La matrice di tracciabilità è uno strumento utilizzato nella gestione dei requisiti per tracciare e documentare le relazioni tra diversi elementi di un sistema. Consente nelle diverse fasi di sviluppo del progetto di verificare che tutti i requisiti siano stati sviluppati e testati.

La matrice verrà aggiornata nelle prossime fasi di sviluppo .

ID REQUISITI	DESIGN	CODICE	TEST	STATO
IF-1.1	BibliotecaService	BibliotecaService.inserisciLibro(Libro) GestioneLibriController.aggiungiAct	BibliotecaServiceTest.testInserisciLibro	FINITO
IF-1.2	BibliotecaService	BibliotecaService.modificaLibro(...) GestioneLibriController.modificaAct	BibliotecaServiceTest.testModificaLibro	FINITO
IF-1.3	BibliotecaService	BibliotecaService.eliminaLibro(String isbn) GestioneLibriController.eliminaAct	BibliotecaServiceTest.testEliminaLibro; testEliminaLibroConPrestitoAttivo	FINITO
IF-1.4	BibliotecaService	BibliotecaService.cercaLibri(..) GestioneLibriController.cercaAct	BibliotecaServiceTest.testCercaLibri	FINITO
IF-2.1	BibliotecaService	BibliotecaService.inserisciStudente(Studente)/ GestioneStudentiController.aggiungiAct	BibliotecaServiceTest.testInserisciStudente	FINITO
IF-2.2	BibliotecaService	BibliotecaService.modificaStudente(...) GestioneStudentiController.modificaAct	BibliotecaServiceTest.testModificaStudente	FINITO
IF-2.3	BibliotecaService	BibliotecaService.eliminaStudente(Stringmatricola) GestioneStudentiController.eliminaAct	BibliotecaServiceTest.testEliminaStudente,testEliminaStudenteConPrestitoAttivo	FINITO
IF-2.4	BibliotecaService	BibliotecaService.cercaStudente(...)	BibliotecaServiceTest	FINITO

		GestioneStudentiController.cercaAct		
IF-3.1	BibliotecaService	BibliotecaService.registraPrestito(...) GestionePrestitiController.registraAct	BibliotecaServiceTest.testRegistraPrestito	FINITO
IF-3.2	BibliotecaService	BibliotecaService.registraPrestito(...)	BibliotecaServiceTest.testRegistraPrestitoLimiteCopie	FINITO
IF-3.3	Studente,BibliotecaService	Studente.aggiungiPrestito(p) BibliotecaService.registraPrestito(...)	StudenteTest.testAggiungiPrestito BibliotecaServiceTest.testRegistraPrestitoLimite3Prestiti	FINITO
IF-3.4	Libro,BibliotecaService	Libro.decrementaCopie()	BibliotecaServiceTest.testRegistraPrestito	FINITO
IF-3.5	BibliotecaService, Prestito	BibliotecaService.elencoPrestitiAttiviOrdinatiPerDataPrevista() Prestito.isInRitardo	BibliotecaServiceTest.testElencoPrestitiAttiviOrdinati PrestitoTest.testIsInRitardo	FINITO
IF-3.6	Libro, BibliotecaService	Libro.incrementaCopie() BibliotecaService.registraRestituzione	BibliotecaServiceTest.testRegistraRestituzione StudenteTest.testChiudiPrestito	FINITO
DF-1.1	Libro	Libro (Costruttore e Getter/Setter)	LibroTest.testCostruttoreValido; testCostruttoreTitoloNullo	FINITO
DF-1.2	BibliotecaService	BibliotecaService.inserisciLibro(...)	BibliotecaServiceTest.testInserisciLibroDuplicato	FINITO
DF-1.3	Libro	Libro.setCopieDisponibili(...)	LibroTest.testSetCopieDisponibiliNegativo	FINITO
DF-1.5	Libro	Libro.setAnnoPubblicazione	LibroTest.testCostruttoreValido	FINITO
DF-2.1	Studente	Studente (Costruttore e Getter/Setter)	StudenteTest.testGettersAndSetters	FINITO
DF-2.2	BibliotecaService	BibliotecaService.inserisciStudente(...)	BibliotecaServiceTest.testInserisciStudenteDuplicato	FINITO