
**SOFTWARE
BIBLIOTECA UNIVERSITARIA**

**Acerra Fabrizio
Affinita Natale
Cwierotka Jakub
Galluccio Hermann**

Novembre 2025

Indice

1	Specifica dei Requisiti del Software - SRS	3
1.1	Requisiti Funzionali	3
1.1.1	Funzionalità individuali	3
1.1.2	Esigenze di dati e informazioni	6
1.1.3	Interfacce con sistemi esterni	7
1.1.4	Interfaccia Utente	7
1.2	Requisiti non funzionali	8
1.3	Tabella di categorizzazione dei requisiti	9
1.4	Casi d'uso	11
1.5	Diagrammi dei casi d'uso	19
2	Design	21
2.1	Principi di buona progettazione	21
2.1.1	Progettazione di dettaglio	21
2.1.2	Principio della minima sorpresa	21
2.1.3	Principio della singola responsabilità	21
2.1.4	Principio aperto/chiuso	22
2.1.5	Principio di segregazione delle interfacce	22
2.1.6	Principio di inversione delle dipendenze	22
2.1.7	Separazione delle preoccupazioni	22
2.1.8	Don't Repeat Yourself (DRY)	23
2.1.9	You Ain't Gonna Need It (YANGI)	23
2.1.10	Gerarchia dei modelli	24
2.1.11	Prospettiva del MainController	25
2.1.12	Prospettiva dei RegisterController	26
2.2	Diagrammi di sequenza	29
2.2.1	Livelli di coesione	35
2.2.2	Livelli di accoppiamento	36

1 Specifica dei Requisiti del Software - SRS

1.1 Requisiti Funzionali

1.1.1 Funzionalità individuali

IF-1 Registrazione di un libro:

Il sistema deve consentire la registrazione di un libro nel catalogo della biblioteca. Durante tale procedura, il sistema deve acquisire tutti i dati definiti nel requisito [DF-1](#) e presentare all'admin l'interfaccia descritta nel requisito [UI-2](#).

IF-2 Modifica dati di un libro registrato:

Il sistema deve consentire la modifica dei dati associati ad un libro precedentemente registrato. Durante tale procedura, il sistema deve acquisire tutti i dati modificabili definiti nel requisito [DF-1](#) e presentare all'admin l'interfaccia descritta nel requisito [UI-3](#).

IF-3 Rimozione di un libro:

Il sistema deve consentire la rimozione di un libro precedentemente registrato dal catalogo libri. Durante tale procedura, il sistema deve presentare all'admin l'interfaccia descritta nel requisito [UI-5](#).

IF-4 Ordinamento libri:

Il sistema deve ordinare lessicograficamente il catalogo dei libri in base al titolo, autori e codice identificativo univoco.

IF-5 Visualizzazione del catalogo libri:

Il sistema deve consentire la visualizzazione del catalogo libri precedentemente registrati. Durante tale procedura, il sistema deve mostrare tutti i dati definiti nel requisito [DF-1](#), ordinarli secondo il requisito [IF-4](#) e presentare all'admin l'interfaccia descritta nel requisito [UI-1](#).

IF-6 Ricerca libro nel catalogo:

Il sistema deve consentire la ricerca di un libro all'interno del catalogo libri sulla base di titolo, autore o codice identificativo univoco.

IF-7 Unicità libro:

Il sistema deve garantire che non siano presenti due libri identici (dunque, aventi lo stesso identificativo secondo [DF-1](#)) nel catalogo libri.

IF-8 Verifica dati libro:

Il sistema deve verificare che tutti i dati inseriti nei campi specificati nel requisito

[DF-1](#) siano conformi al formato previsto. Non deve essere consentita la presenza nel sistema di un libro i cui dati non rispettino tale formato.

IF-9 Registrazione di uno studente:

Il sistema deve consentire la registrazione di uno studente nel registro della biblioteca. Durante tale procedura, il sistema deve acquisire tutti i dati definiti nel requisito [DF-2](#), fatta eccezione per "Lista dei libri attualmente in prestito", e presentare all'admin l'interfaccia descritta nel requisito [UI-2](#).

IF-10 Modifica di uno studente:

Il sistema deve consentire la modifica dei dati associati ad uno studente precedentemente registrato. Durante tale procedura, il sistema deve acquisire tutti i dati modificabili definiti nel requisito [DF-2](#), fatta eccezione per "Lista dei libri attualmente in prestito", e presentare all'admin l'interfaccia descritta nel requisito [UI-3](#).

IF-11 Rimozione di uno studente:

Il sistema deve consentire la rimozione di uno studente precedentemente registrato dal catalogo studenti. Durante tale procedura, il sistema deve presentare all'admin l'interfaccia descritta nel requisito [UI-5](#).

IF-12 Ordinamento studenti:

Il sistema deve ordinare lessicograficamente il registro degli studenti in base a cognome, nome e matricola.

IF-13 Visualizzazione del registro studenti:

Il sistema deve consentire la visualizzazione del registro studenti precedentemente registrati. Durante tale procedura, il sistema deve mostrare tutti i dati definiti nel requisito [DF-2](#), fatta eccezione per "Lista dei libri attualmente in prestito", e presentare all'admin l'interfaccia descritta nel requisito [UI-1](#).

IF-14 Ricerca studente nel catalogo:

Il sistema deve consentire la ricerca di uno studente all'interno del registro sulla base di cognome, nome o matricola

IF-15 Unicità studente:

Il sistema deve garantire che non siano presenti due studenti identici (dunque, aventi lo stesso identificativo secondo [DF-2](#)) nel registro studenti.

IF-16 Verifica dati studente:

Il sistema deve verificare che tutti i dati inseriti nei campi specificati nel requisito [DF-2](#) siano conformi al formato previsto. Non deve essere consentita la presenza nel sistema di uno studente i cui dati non rispettino tale formato.

IF-17 Registrazione di un prestito:

Il sistema deve consentire la registrazione di un prestito nel registro della biblioteca. Durante tale procedura, il sistema deve acquisire tutti i dati definiti nel re-

quisito [DF-3](#), fatta eccezione per "Restituzione avvenuta o in attesa", e presentare all'admin l'interfaccia descritta nel requisito [UI-1](#).

IF-18 Modifica di un prestito:

Il sistema deve consentire la modifica del dato "Data ultima di restituzione" di un prestito precedentemente registrato. Durante tale procedura, il sistema deve acquisire il dato specificato nel requisito [DF-3](#) e presentare all'admin l'interfaccia descritta nel requisito [UI-3](#).

IF-19 Ordinamento prestiti:

Il sistema deve ordinare cronologicamente l'elenco di prestiti in base alla data di ultima restituzione.

IF-20 Visualizzazione del catalogo prestiti:

Il sistema deve consentire la visualizzazione del catalogo prestiti precedentemente registrati. Durante tale procedura, il sistema deve mostrare tutti i dati definiti nel requisito [DF-3](#) e presentare all'admin l'interfaccia descritta nel requisito [UI-1](#).

IF-21 Evidenziazione prestiti in ritardo:

Al momento della visualizzazione dei prestiti, il sistema deve evidenziare tutti i prestiti in ritardo per facilitarne l'individuazione da parte dell'admin.

IF-22 Registrazione della restituzione di un prestito:

Il sistema deve consentire la registrazione della restituzione di un prestito precedentemente registrato. Durante tale procedura, il sistema deve presentare all'admin l'interfaccia descritta nel requisito [UI-5](#).

IF-23 Rimozione di un prestito:

Il sistema deve consentire la rimozione di un prestito precedentemente registrato dal catalogo prestiti. Se il prestito non è stato ancora estinto il sistema lo farà in automatico. Durante tale procedura, il sistema deve presentare all'admin l'interfaccia descritta nel requisito [UI-5](#).

IF-24 Verifica dati prestito:

Il sistema deve verificare che tutti i dati inseriti nei campi specificati nel requisito [DF-3](#) siano conformi al formato previsto. Non deve essere consentita la presenza nel sistema di un prestito i cui dati non rispettino tale formato.

IF-25 Verifica limite massimo prestiti:

Il sistema deve garantire che ad un determinato studente non possano essere associati più di 3 prestiti contemporaneamente.

IF-26 Verifica presenza libro:

Il sistema non deve consentire la registrazione di un prestito il cui libro associato sia stato momentaneamente esaurito.

IF-27 Aggiornamento copie disponibili di un libro:

Il sistema deve automaticamente decrementare il numero di copie di un libro nel momento in cui questo viene associato ad un prestito. Analogamente, il sistema deve incrementare il numero di copie di un libro precedentemente prestato nel momento in cui esso viene restituito.

IF-28 Salvataggio archivio:

Il sistema deve consentire il salvataggio su file dell'intero database della biblioteca, che comprende: catalogo dei libri, registro degli studenti e registro prestiti.

IF-29 Caricamento archivio:

Il sistema deve consentire l'apertura di un file di salvataggio per caricare il database della biblioteca che comprende: catalogo dei libri, registro degli studenti e registro prestiti in esso contenuti.

1.1.2 Esigenze di dati e informazioni

DF-1 Dati di un libro:

Il sistema deve memorizzare, per ogni libro registrato, i seguenti dati:

- Titolo (*modificabile*).
- Autori (nome e cognome) (*modificabile*).
- Anno di pubblicazione (*modificabile*).
- Codice identificativo univoco (*identificativo*).
- Numero di copie disponibili (*modificabile*).

DF-2 Dati di uno studente;

Il sistema deve memorizzare, per ogni studente registrato, i seguenti dati:

- Nome (*modificabile*).
- Cognome (*modificabile*).
- Matricola (*identificativo*).
- E-Mail istituzionale (termina in @studenti.unisa.it) (*modificabile*).
- Lista dei libri attualmente in prestito (*modificabile*).

DF-3 Dati di un prestito:

Il sistema deve memorizzare, per ogni prestito registrato, i seguenti dati:

- Studente a cui è associato il prestito.
- Libro preso in prestito.
- Data ultima di restituzione (*modificabile*)
- Restituzione avvenuta o in attesa (*modificabile*)

1.1.3 Interfacce con sistemi esterni

IS-1 Memorizzazione archivio su file:

Il sistema deve salvare su file tutte le informazioni circa:

- Catalogo dei libri.
- Registro degli studenti.
- Registro dei prestiti.

1.1.4 Interfaccia Utente

UI-1 Schermata principale. Deve contenere:

- Un modo per cambiare la tipologia di informazioni da visualizzare.
- Una visualizzazione delle informazioni in forma tabulare.
- Un modo per salvare e caricare l'archivio.
- Un modo per aggiungere le informazioni della tipologia selezionata [UI-2](#).
- Un modo per modificare le informazioni della tipologia selezionata [UI-3](#).
- Un modo per eliminare le informazioni della tipologia selezionata [UI-5](#).
- Un modo per cercare le informazioni della tipologia selezionata.

UI-2 Schermata di inserimento dati. Deve contenere:

- Dei controlli per inserire i dati per la tipologia di informazione selezionata.
- Un modo per annullare l'inserimento.
- Un modo per confermare l'inserimento.

UI-3 Schermata di modifica dati. Deve contenere:

- Dei controlli per inserire i dati modificabili per la tipologia di informazione selezionata.
- Dei controlli in sola lettura per le informazioni non modificabili per la tipologia di informazione selezionata.
- Un modo per annullare la modifica.
- Un modo per confermare la modifica.

UI-4 Pop-Up di notifica. Deve contenere:

- Un messaggio di notifica.
- Un modo per confermare e chiudere la notifica.

UI-5 Pop-Up di conferma. Deve contenere:

- Un'avvertenza.
- Un modo per confermare l'operazione.
- Un modo per annullare l'operazione.

1.2 Requisiti non funzionali

FC-1 Portabilità:

Il software deve essere progettato per poter essere eseguito sui sistemi operativi Windows, MacOS e Linux.

FC-2 Interfaccia responsive:

Il software ha un'interfaccia che si adatta a varie dimensioni di schermo.

1.3 Tabella di categorizzazione dei requisiti

Prefisso	Requisito	Priorità	Categoria
IF-1	Registrazione di un libro	Alta	Funzionalità Individuale
IF-2	Modifica dati di un libro	Media	Funzionalità Individuale
IF-3	Rimozione di un libro	Alta	Funzionalità Individuale
IF-4	Ordinamento libri	Bassa	Funzionalità Individuale
IF-5	Visualizzazione catalogo libri	Alta	Funzionalità Individuale
IF-6	Ricerca libro nel catalogo	Media	Funzionalità Individuale
IF-7	Unicità libro	Alta	Funzionalità Individuale
IF-8	Verifica dati libro	Alta	Funzionalità Individuale
IF-9	Registrazione di uno studente	Alta	Funzionalità Individuale
IF-10	Modifica di uno studente	Media	Funzionalità Individuale
IF-11	Rimozione di uno studente	Alta	Funzionalità Individuale
IF-12	Ordinamento studenti	Bassa	Funzionalità Individuale
IF-13	Visualizzazione del registro studenti	Alta	Funzionalità Individuale
IF-14	Ricerca studente nel catalogo	Media	Funzionalità Individuale
IF-15	Unicità studente	Alta	Funzionalità Individuale
IF-16	Verifica dati studente	Alta	Funzionalità Individuale
IF-17	Registrazione di un prestito	Alta	Funzionalità Individuale
IF-18	Modifica di un prestito	Media	Funzionalità Individuale
IF-19	Ordinamento prestiti	Bassa	Funzionalità Individuale
IF-20	Visualizzazione catalogo prestiti	Alta	Funzionalità Individuale
IF-21	Evidenziazione prestiti in ritardo	Bassa	Funzionalità Individuale
IF-22	Registrazione della restituzione di un prestito	Media	Funzionalità Individuale
IF-23	Rimozione di un prestito	Bassa	Funzionalità Individuale
IF-24	Verifica dati prestito	Alta	Funzionalità Individuale
IF-25	Verifica limite massimo prestiti	Alta	Funzionalità Individuale
IF-26	Verifica presenza libro	Alta	Funzionalità Individuale
IF-27	Aggiornamento copie disponibili	Alta	Funzionalità Individuale

Prefisso	Requisito	Priorità	Categoria
IF-28	Salvataggio archivio	Media	Funzionalità Individuale
IF-29	Caricamento archivio	Media	Funzionalità Individuale
DF-1	Dati di un libro	Alta	Esigenze dati e informazioni
DF-2	Dati di uno studente	Alta	Esigenze dati e informazioni
DF-3	Dati di un prestito	Alta	Esigenze dati e informazioni
IS-1	Memorizzazione archivio su file	Media	Interfaccia con Sistemi esterni
UI-1	Schermata principale	Alta	Interfaccia Utente
UI-2	Schermata di inserimento dati	Alta	Interfaccia Utente
UI-3	Schermata di modifica dati	Media	Interfaccia Utente
UI-4	Pop-up di notifica	Bassa	Interfaccia Utente
UI-5	Pop-up di conferma	Bassa	Interfaccia Utente
FC-2	Portabilità	Alta	Requisiti non funzionali
FC-2	Interfaccia responsive	Bassa	Requisiti non funzionali

1.4 Casi d'uso

UC-1 Titolo: Registrazione libro

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "libro".

Flusso principale:

1. L'admin avvia la funzione di inserimento.
2. L'admin compila i campi specificati in [DF-1](#).
3. L'admin conferma l'operazione

Post-condizioni: Il libro è correttamente inserito nell'elenco libri.

Flusso alternativo 1:

- 4 I dati inseriti sono errati. L'admin deve inserirne di nuovi. Si riparte dal punto 2.

Flusso alternativo 2:

- 3 L'admin annulla l'operazione o chiude la finestra.

Flusso alternativo 3:

- 4 È già presente un libro avente lo stesso identificativo. Si riparte dal punto 2.
-

UC-2 Titolo: Modifica libro

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "libro".

Flusso principale:

1. L'admin seleziona un libro da modificare.
2. L'admin avvia la funzione di modifica.
3. L'admin compila i campi (modificabili) specificati in [DF-1](#).
4. L'admin conferma l'operazione.

Post-condizioni: Il libro è correttamente modificato.

Flusso alternativo 1:

- 5 I dati inseriti sono errati. L'admin deve inserire di nuovi. Si riparte dal punto 2.

Flusso alternativo 2:

3 L'admin annulla l'operazione o chiude la finestra.

UC-3 Titolo: Rimozione libro

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "libro".

Flusso principale:

1. L'admin seleziona un libro da rimuovere.
2. L'admin avvia la funzione di rimozione.
3. L'admin conferma la rimozione.

Post-condizioni: Il libro è correttamente rimosso dall'elenco libri.

Flusso alternativo 1:

- 3 L'admin annulla l'operazione o chiude la finestra
-

UC-4 Titolo: Visualizzazione libri

Attori: Admin

Precondizioni: L'admin si trova in una schermata di visualizzazione.

Flusso principale:

1. L'admin accede alla schermata di visualizzazione selezionando la tipologia di elemento "libro".
2. L'admin scorre i libri della schermata di visualizzazione.

Post-condizioni: L'admin visualizza i libri registrati

UC-5 Titolo: Ricerca titolo

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "libro".

Flusso principale:

1. L'admin scrive del testo all'interno della barra di ricerca

Post-condizioni: Il sistema mostra dei risultati sulla base delle corrispondenze trovate rispetto al testo inserito nella barra di ricerca.

UC-6 Titolo: Registrazione studente

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "studente".

Flusso principale:

1. L'admin avvia la funzione di inserimento.
2. L'admin compila i campi specificati in [DF-2](#), eccetto "Lista dei libri attualmente in prestito".
3. L'admin conferma l'operazione.

Post-condizioni: Lo studente è correttamente inserito nell'elenco studenti.

Flusso alternativo 1:

- 3 I dati inseriti sono errati. L'admin deve inserirne di nuovi. Si riparte dal punto 2.

Flusso alternativo 2:

- 3 L'admin annulla l'operazione o chiude la finestra.

Flusso alternativo 3:

- 4 È già presente uno studente avente stesso identificativo. Si riparte dal punto 2.
-

UC-7 Titolo: Modifica studente

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "studente".

Flusso principale:

1. L'admin seleziona uno studente da modificare.
2. L'admin avvia la funzione di modifica.
3. L'admin compila i campi (modificabili) specificati in [DF-2](#), eccetto "Lista dei libri attualmente in prestito".
4. L'admin conferma l'operazione

Post-condizioni: Lo studente è correttamente modificato.

Flusso alternativo 1:

- 5 I dati sono errati. L'admin deve inserirne di nuovi. Si riparte dal punto 2.

Flusso alternativo 2:

4 L'admin annulla l'operazione o chiude la finestra.

UC-8 Titolo: Rimozione studente

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "studente".

Flusso principale:

1. L'admin seleziona uno studente da rimuovere.
2. L'admin avvia la funzione di rimozione
3. L'admin conferma la rimozione

Post-condizioni: Lo studente è correttamente rimosso dall'elenco studenti

Flusso alternativo 1:

- 4 L'admin annulla l'operazione o chiude la finestra.
-

UC-9 Titolo: Visualizzazione studenti

Attori: Admin

Precondizioni: L'admin si trova in una schermata di visualizzazione.

Flusso principale:

1. L'admin accede alla schermata di visualizzazione.
2. L'admin scorre gli studenti nella schermata di visualizzazione

Post-condizioni: L'admin visualizza gli studenti registrati.

UC-10 Titolo: Ricerca studente

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "studente".

Flusso principale:

1. L'admin scrive del testo all'interno della barra di ricerca.

Post-condizioni: Il sistema mostra dei risultati sulla base delle corrispondenze trovate rispetto al testo inserito nella barra di ricerca.

UC-11 Titolo: Registrazione prestito

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "prestito".

Flusso principale:

1. L'admin avvia la funzione di inserimento.
2. L'admin compila i campi specificati in [DF-3](#), eccetto "Restituzione avvenuta o in attesa".
3. L'admin conferma l'operazione.

Post-condizioni: Il prestito è correttamente inserito nell'elenco prestiti.

Flusso alternativo 1:

- 4 I dati inseriti sono errati. L'admin deve inserirne di nuovi. Si riparte dal punto 2.

Flusso alternativo 2:

- 3 L'admin annulla l'operazione o chiude la finestra.

Flusso alternativo 3:

- 4 Lo studente specificato ha già 3 prestiti non ancora estinti. Si riparte dal punto 2.

Flusso alternativo 4:

- 4 Il libro specificato è esaurito (nessuna copia rimanente). Si riparte dal punto 2.

UC-12 Titolo: Modifica prestito

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "prestito".

Flusso principale:

1. L'admin seleziona un prestito da modificare.
2. L'admin avvia la funzione di modifica.
3. L'admin modifica "Data ultima di restituzione"
4. L'admin conferma l'operazione.

Post-condizioni: Il prestito è correttamente modificato.

Flusso alternativo 1:

- 5 I dati inseriti sono errati. L'admin deve inserirne di nuovi. Si riparte dal punto 2.

Flusso alternativo 2:

- 5 L'admin annulla l'operazione o chiude la finestra.
-

UC-13 Titolo: Rimozione prestito

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "prestito".

Flusso principale:

1. L'admin seleziona un prestito da rimuovere.
2. L'admin avvia la funzione di rimozione.
3. L'admin conferma la rimozione.

Post-condizioni: Il prestito è correttamente rimosso dall'elenco prestiti.

Flusso alternativo 1:

- 3 L'admin annulla l'operazione o chiude la finestra.
-

UC-14 Titolo: Visualizzazione prestiti

Attori: Admin

Precondizioni: L'admin si trova nella schermata di una visualizzazione.

Flusso principale:

1. L'admin accede alla schermata di visualizzazione.
2. L'admin scorre i prestiti nella schermata di visualizzazione

Post-condizioni: L'admin visualizza i prestiti registrati.

UC-15 Titolo: Ricerca prestito

Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "prestito".

Flusso principale:

1. L'admin scrive del testo all'interno della barra di ricerca.

Post-condizioni: Il sistema mostra dei risultati sulla base delle corrispondenze trovate rispetto al testo inserito nella barra di ricerca.

UC-16 Titolo: Estinzione prestito
Attori: Admin

Precondizioni: L'admin si trova nella schermata di visualizzazione della tipologia di elemento "prestito".

Flusso principale:

1. L'admin seleziona un prestito da estinguere.
2. L'admin avvia la funzione di estinzione.
3. L'admin conferma l'operazione.

Post-condizioni: Il prestito risulta correttamente contrassegnato come estinto.

Flusso alternativo 1:

- 3 L'admin annulla l'operazione o chiude la finestra.

Flusso alternativo 2:

- 2 Il prestito è già stato estinto. L'operazione è disabilitata.
-

UC-17 Titolo: Salvataggio archivio
Attori: Admin

Precondizioni: L'admin si trova in una schermata di visualizzazione.

Flusso principale:

1. L'admin accede alla schermata di salvataggio.
2. L'admin avvia l'operazione di salvataggio.
3. L'admin conferma l'operazione.

Post-condizioni: L'archivio è correttamente salvato su file.

Flusso alternativo 1:

- 1 L'admin annulla l'operazione o chiude la finestra.
-

UC-18 Titolo: Caricamento archivio
Attori: Admin

Precondizioni: L'admin si trova in una schermata di visualizzazione.

Flusso principale:

1. L'admin accede alla schermata di caricamento file
2. L'admin seleziona il file contenente la biblioteca da caricare.
3. L'admin conferma l'operazione

Post-condizioni: La biblioteca i cui dati erano memorizzati nel file selezionato viene correttamente caricata.

Flusso alternativo 1:

- 4 L'estensione del file selezionato è diversa da quella prevista. L'operazione è annullata: si ritorna al punto 1.

Flusso alternativo 2:

- 4 Il formato del file selezionato è diverso da quello previsto. L'operazione è annullata si ritorna al punto 1.

Flusso alternativo 3:

- 3 L'admin annulla l'operazione o chiude la finestra.
-

1.5 Diagrammi dei casi d'uso

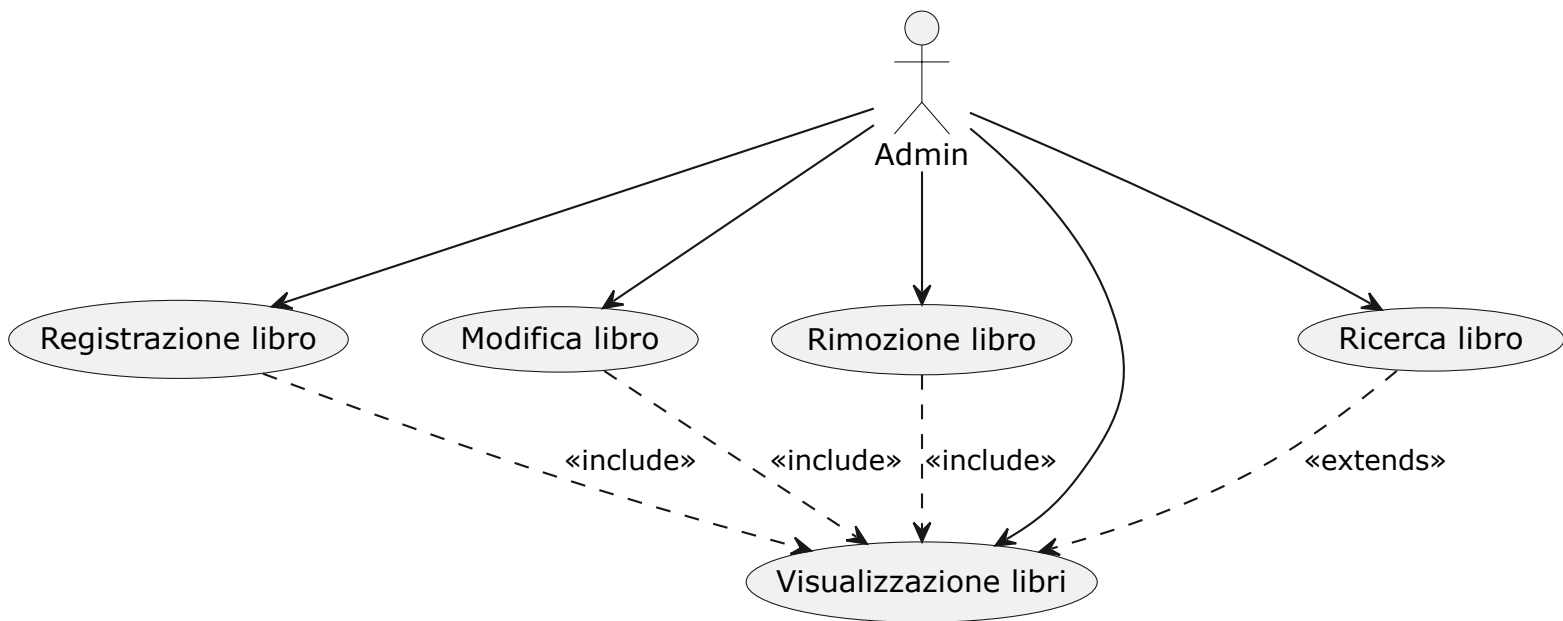


Figura 1.1: Diagramma dei casi d'uso: gestione dei libri (UC-1, UC-2, UC-3, UC-4, UC-5).

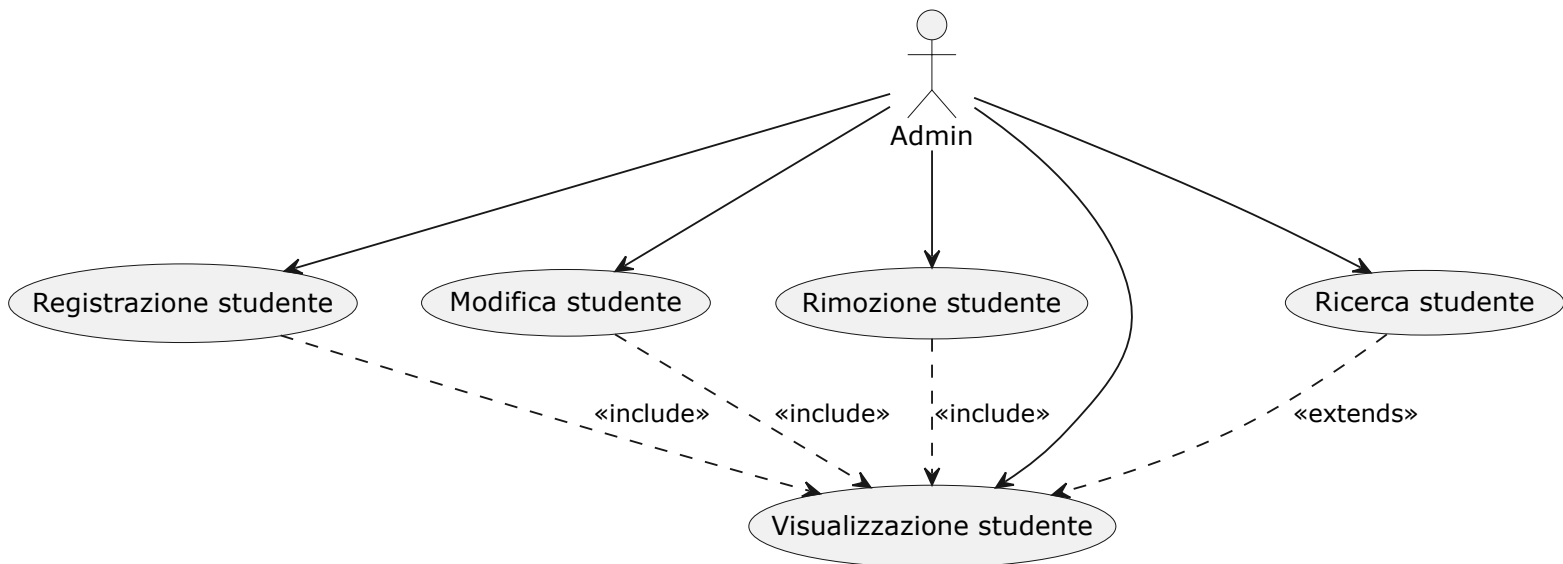


Figura 1.2: Diagramma dei casi d'uso: gestione degli studenti (UC-6, UC-7, UC-8, UC-9, UC-10).

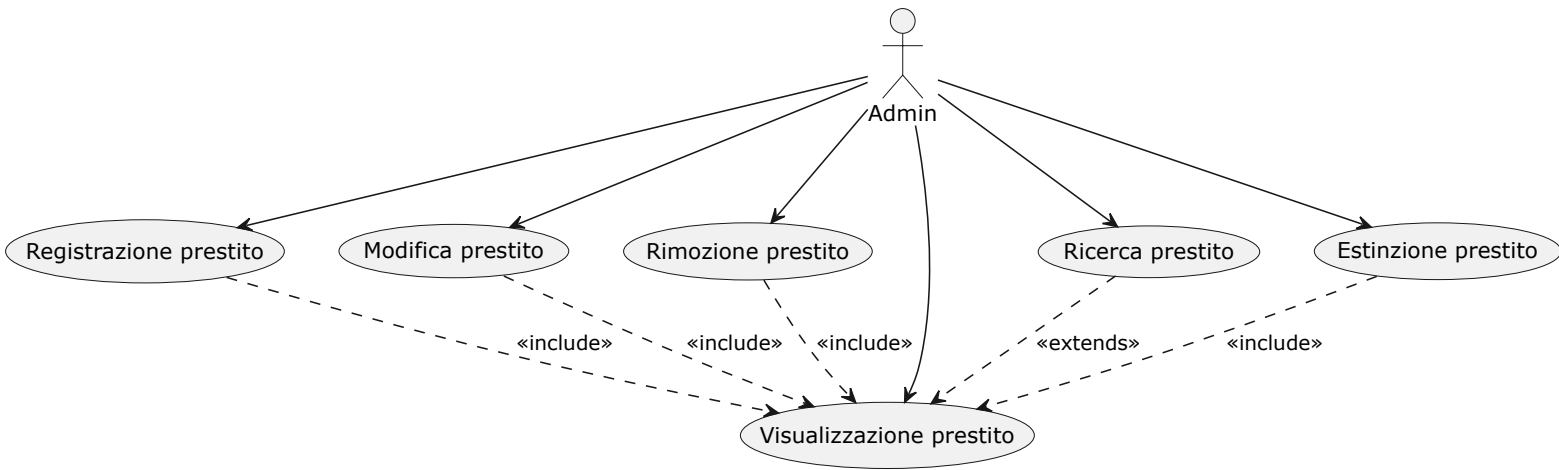


Figura 1.3: Diagramma dei casi d'uso: gestione dei prestiti (UC-11, UC-12, UC-13, UC-14, UC-15, UC-16).

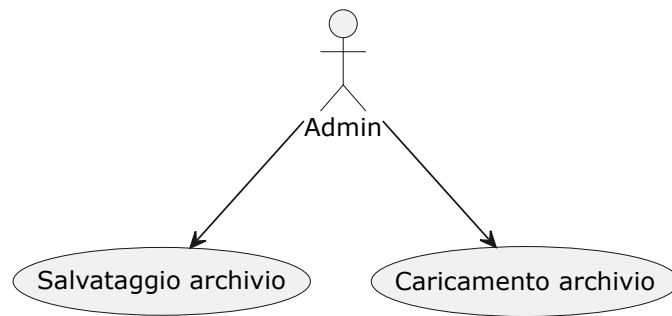


Figura 1.4: Diagramma dei casi d'uso: gestione dell'I/O (UC-17, UC-18).

2 Design

2.1 Principi di buona progettazione

2.1.1 Progettazione di dettaglio

È stata applicata una decomposizione **Object-Oriented** attorno le entità del dominio (*Book*, *Student*, *Loan*). Ogni modulo è concepito come un oggetto che incapsula sia lo stato che i comportamenti specifici, garantendo un'alta coesione interna. L'architettura sfrutta meccanismi di astrazione come *Interfacce*, *Generics* ed *Ereditarietà*, permettendo di definire logiche di gestione comuni che vengono specializzate per ogni tipo di dato (evidenti nell'interfaccia `Register<T>`).

2.1.2 Principio della minima sorpresa

Il codice è stato progettato per garantire intuitività e prevedibilità attraverso l'adozione di una *Naming Convention* coerente con gli standard Java, privilegiando identificatori che, sebbene talvolta estesi, risultano autoesplicativi nel definire scopo e responsabilità di classi, metodi e variabili, minimizzando così le ambiguità e facilitando l'immediata comprensione del sistema.

2.1.3 Principio della singola responsabilità

Il principio è stato applicato a tutti i livelli dell'architettura per garantire alta coesione:

- **Modelli:** Le classi *Student*, *Book* e *Loan* sono **POJO** (*Plain Old Java Objects*) che hanno l'unica responsabilità di rappresentare i dati e garantirne la validità tramite controlli interni, senza contenere logica di business complessa o riferimenti alla GUI.
- **IOManager:** La gestione della persistenza è isolata nella classe `LibraryIOManager`, che si occupa esclusivamente di serializzazione e deserializzazione, rimanendo totalmente disaccoppiata dalla logica operativa della biblioteca.
- **Controllers:** La gestione dell'interfaccia è suddivisa per contesti. I `RegisterControllers` (*Student*, *Book*, *Loan*) gestiscono autonomamente i propri tab, mentre i `PopupControllers` si occupano esclusivamente degli eventi di inserimento/modifica, separando queste operazioni dalla visualizzazione tabellare principale.

2.1.4 Principio aperto/chiuso

Il sistema è progettato per essere estendibile senza dover modificare il codice esistente:

- **Gerarchia dei Popup:** L'uso di classi astratte (`BookPopupController`, `StudentPopupController`, `LoanPopupController`) definisce una struttura base comune. Se in futuro fosse necessario implementare nuove tipologie di popup (es. per la sola visualizzazione dettagli), basterebbe estendere la classe base senza alterare i controller già funzionanti.

2.1.5 Principio di segregazione delle interfacce

Per evitare dipendenze inutili, è stato evitato il passaggio di oggetti "pesanti" ai componenti che necessitano solo di funzionalità limitate:

- **Gestione Popup:** Invece di passare l'intero oggetto `Register` ai popup di inserimento (dando accesso a metodi non necessari come la rimozione o la lettura), sono state definite interfacce funzionali specifiche e minimali (`RegisterAdder`, `RegisterModifier`). Questo riduce l'accoppiamento e aumenta la sicurezza del codice.
- **Gestione I/O:** La comunicazione tra il `sideBarController` e il `MainController` avviene tramite interfacce `Refresh` per

2.1.6 Principio di inversione delle dipendenze

I moduli di alto livello non dipendono dai dettagli implementativi di basso livello:

- **Astrazione dei Registri:** I controller interagiscono con l'interfaccia generica `Register<T>` e non con le implementazioni concrete (es. `StudentRegister`). Questo permette di cambiare l'implementazione sottostante della struttura dati senza impattare sulla logica dell'interfaccia utente.

2.1.7 Separazione delle preoccupazioni

Il sistema segue una netta divisione logica:

- **Pattern MVC:** L'architettura permette una chiara separazione delle preoccupazioni: i **Controllers** assumono un ruolo di mediazione: il loro compito è limitato esclusivamente alla gestione dell'interazione tra l'utente e l'interfaccia grafica (**View**), recependo gli input e invocando le opportune funzioni dei **Models** senza eseguire logica di business.
- **Gestione I/O:** La logica di come salvare i dati è fisicamente separata dalla logica di cosa salvare, mantenendo il core dell'applicazione pulito da dettagli tecnici di I/O.

2.1.8 Don't Repeat Yourself (DRY)

La duplicazione del codice è stata minimizzata tramite l'ereditarietà:

- **Logica condivisa nei Controller:** Le operazioni comuni, come la chiusura della finestra o gestione della UI, sono implementate una sola volta nelle classi astratte dei popup e riutilizzate automaticamente dalle sottoclassi di inserimento e modifica.

2.1.9 You Ain't Gonna Need It (YANGI)

Lo sviluppo si è attenuto strettamente ai [Requisiti Funzionali](#) evitando l'introduzione di funzionalità aggiuntive non richieste.

2.1.10 Gerarchia dei modelli

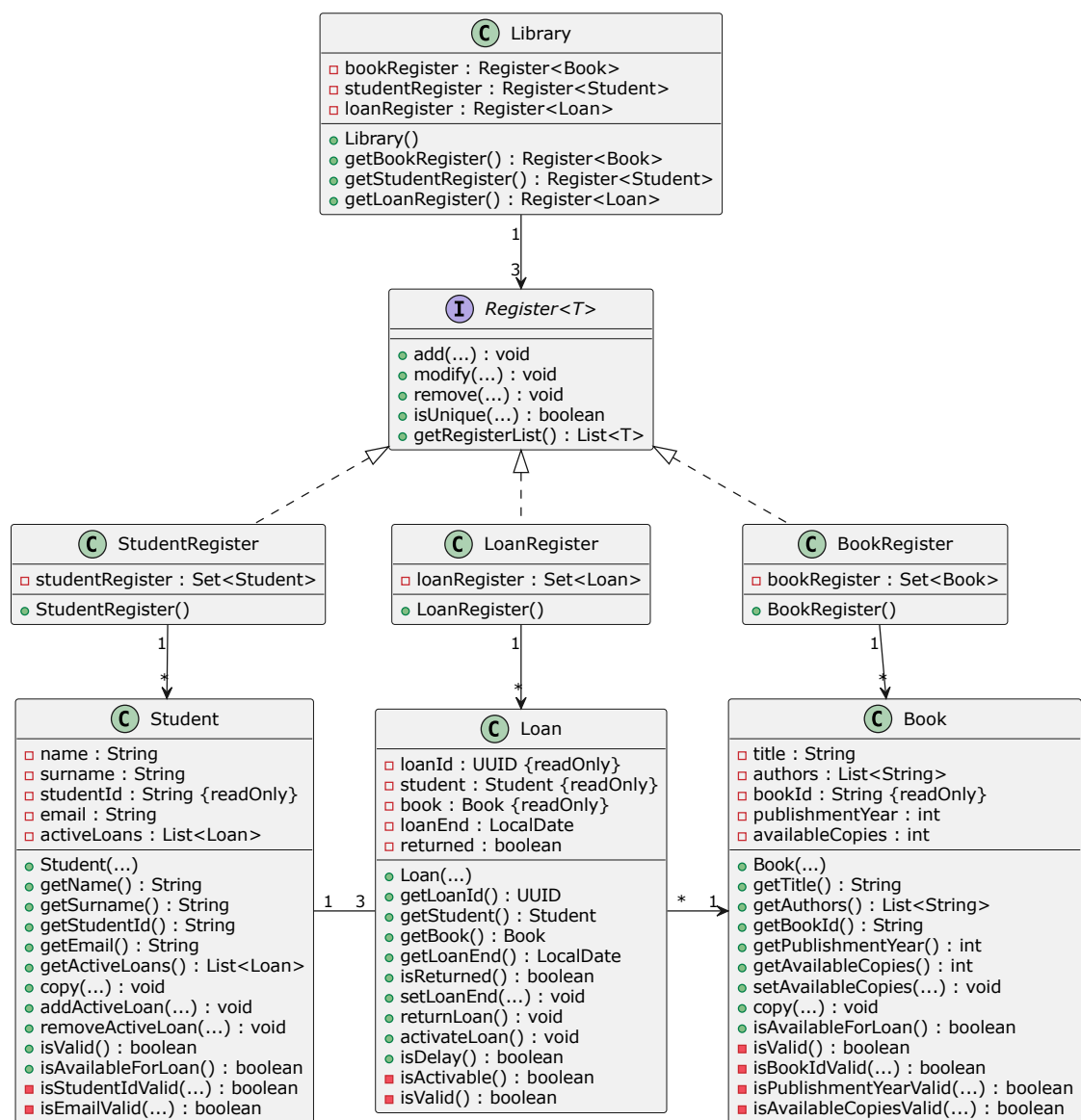


Figura 2.1: Diagramma delle classi per la gerarchia dei modelli.

2.1.11 Prospettiva del MainController

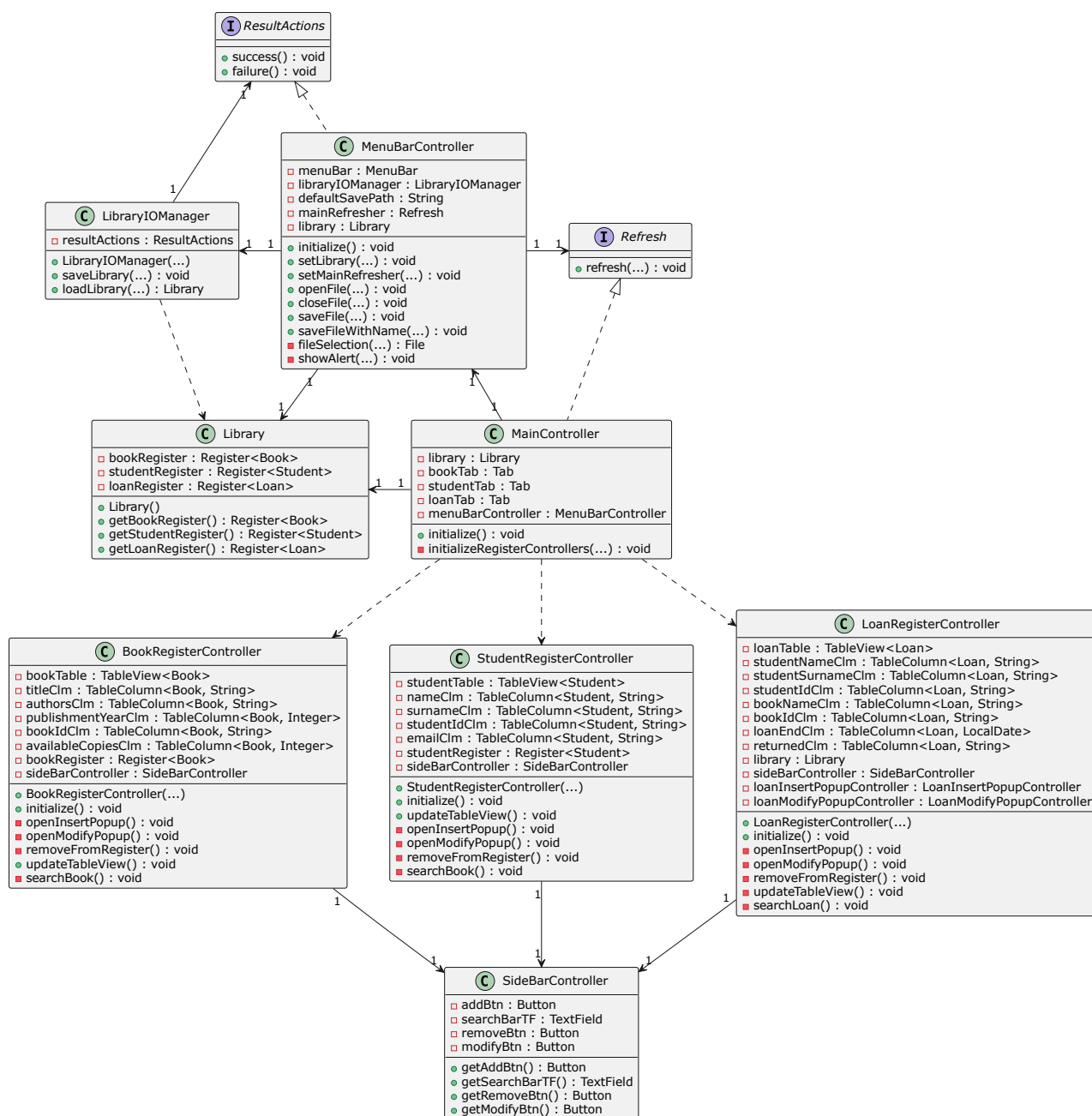


Figura 2.2: Diagramma delle classi per il MainController

2.1.12 Prospettiva dei RegisterController

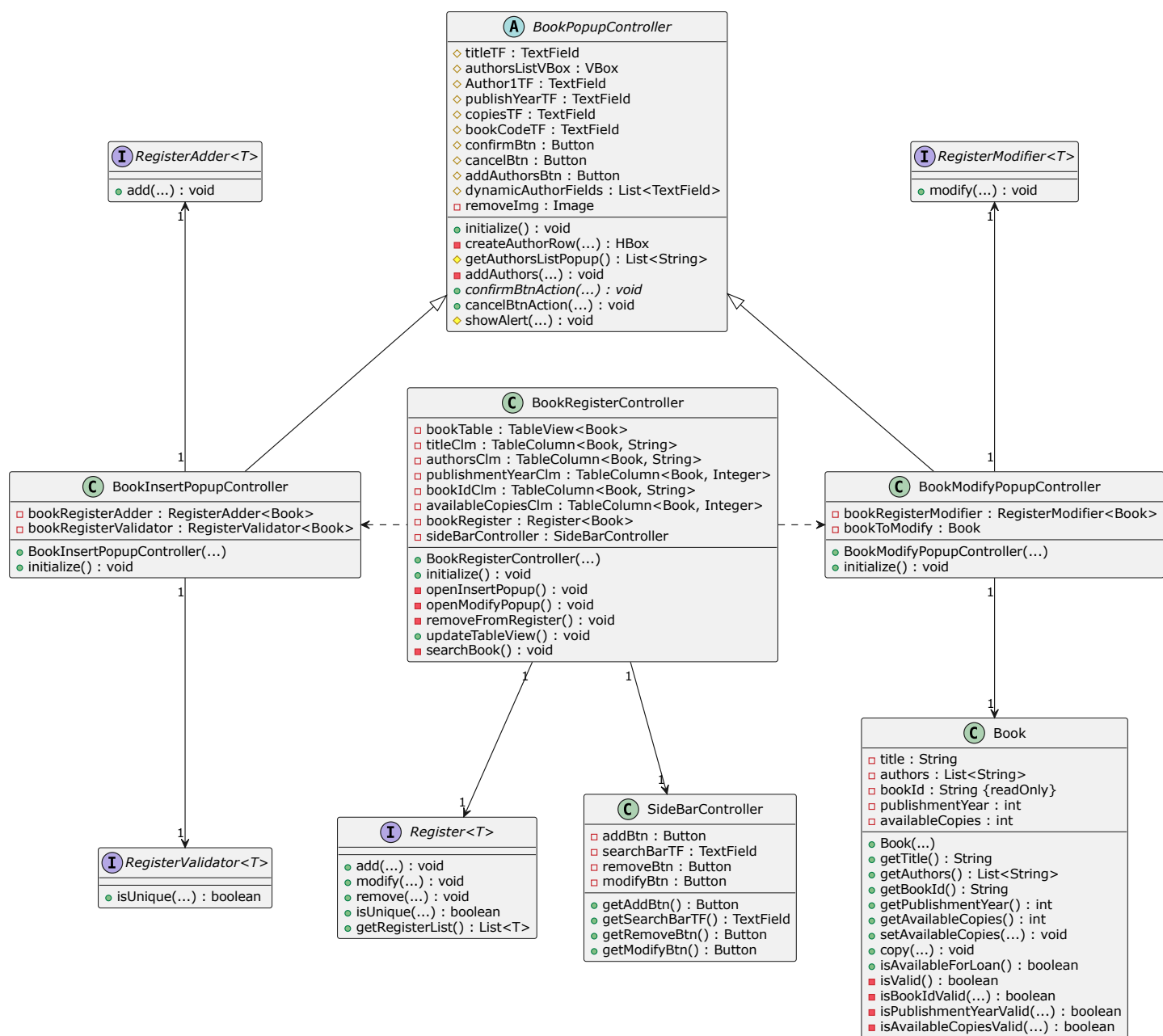


Figura 2.3: Diagramma delle classi dalla prospettiva del BookRegisterController

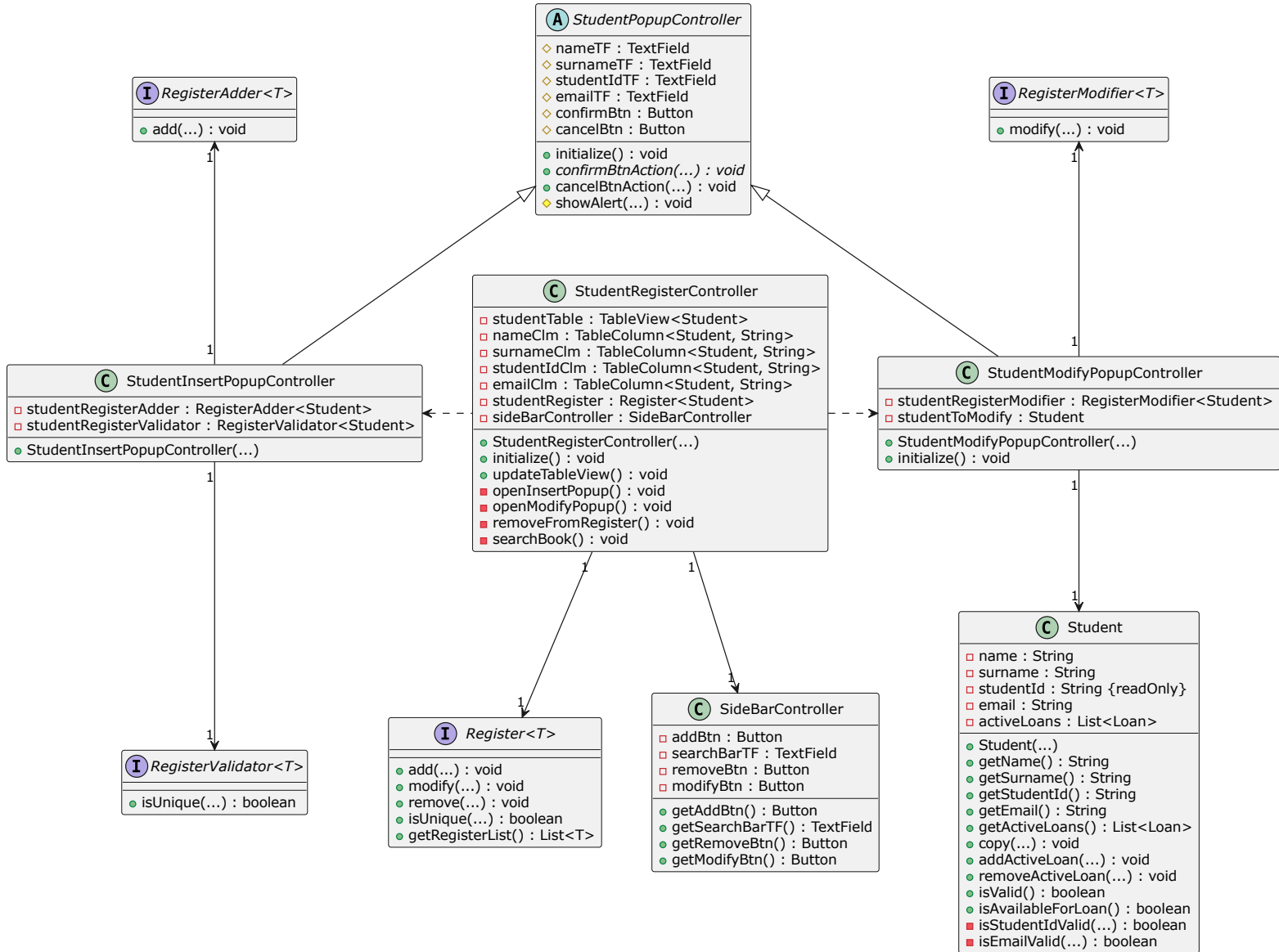


Figura 2.4: Diagramma delle classi dalla prospettiva del StudentRegisterController

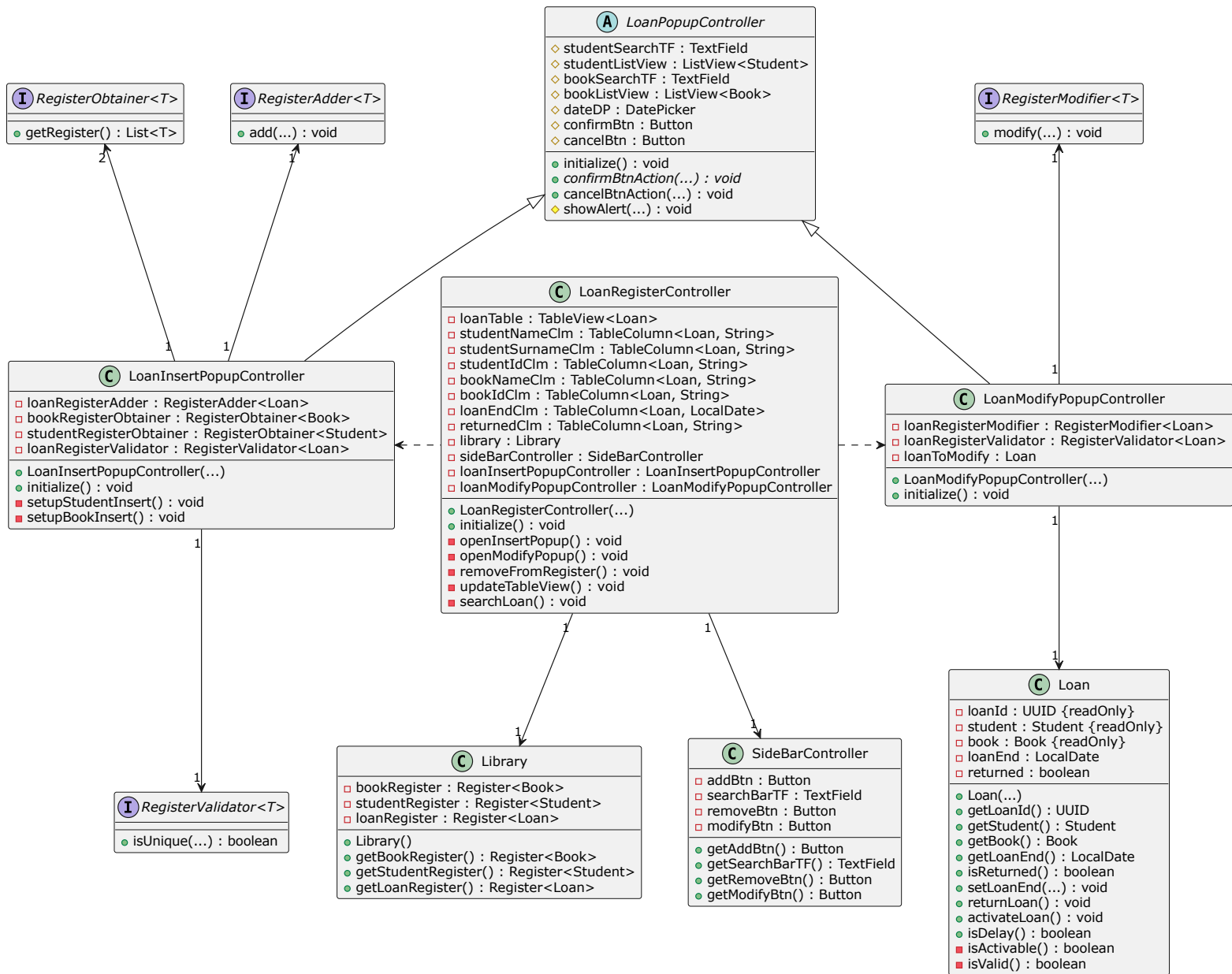


Figura 2.5: Diagramma delle classi dalla prospettiva del LoanRegisterController

2.2 Diagrammi di sequenza

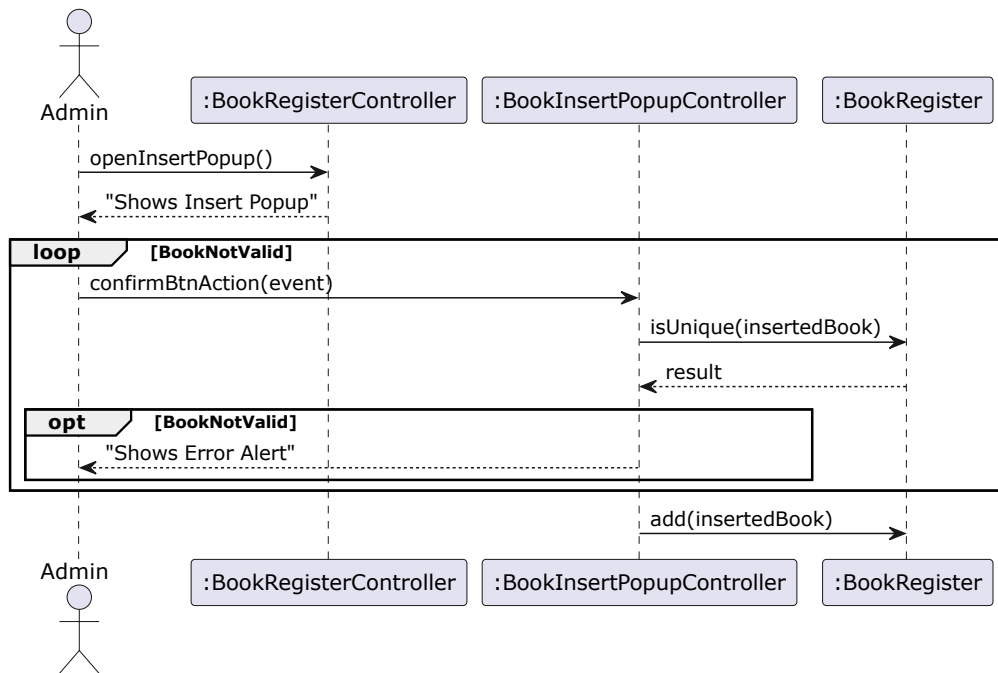


Figura 2.6: Diagramma di sequenza per la registrazione di un libro UC-1

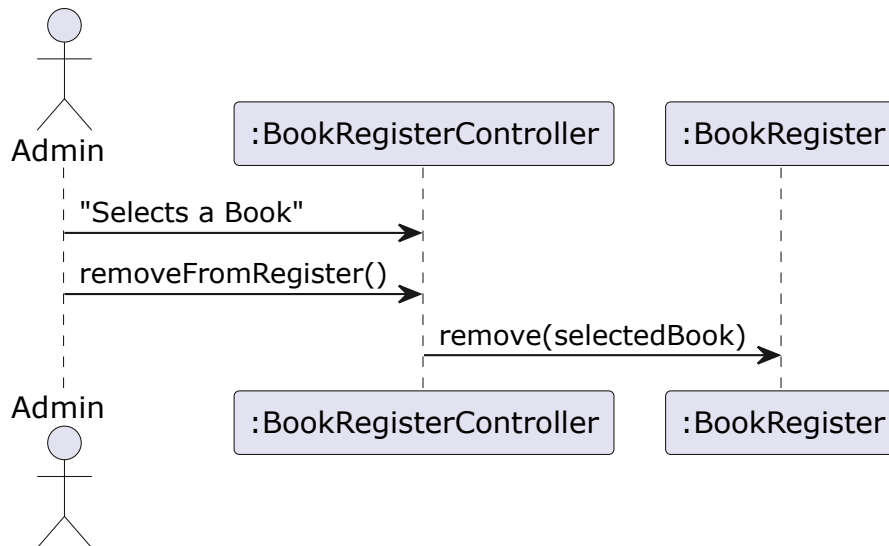


Figura 2.7: Diagramma di sequenza per la rimozione di un libro UC-3

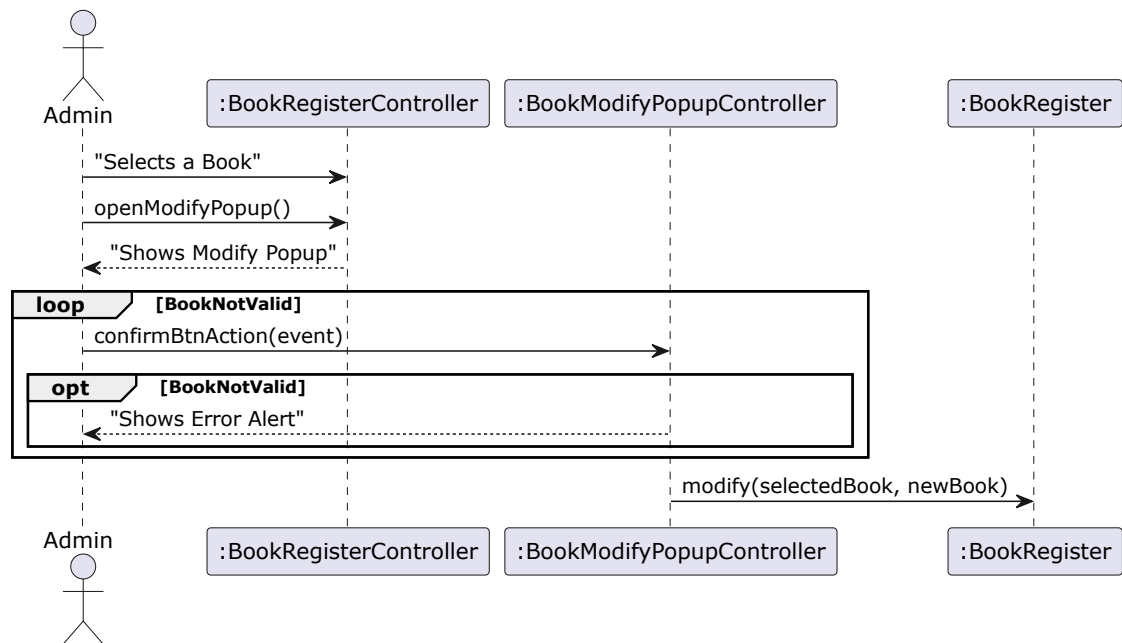


Figura 2.8: Diagramma di sequenza per la modifica di un libro UC-2

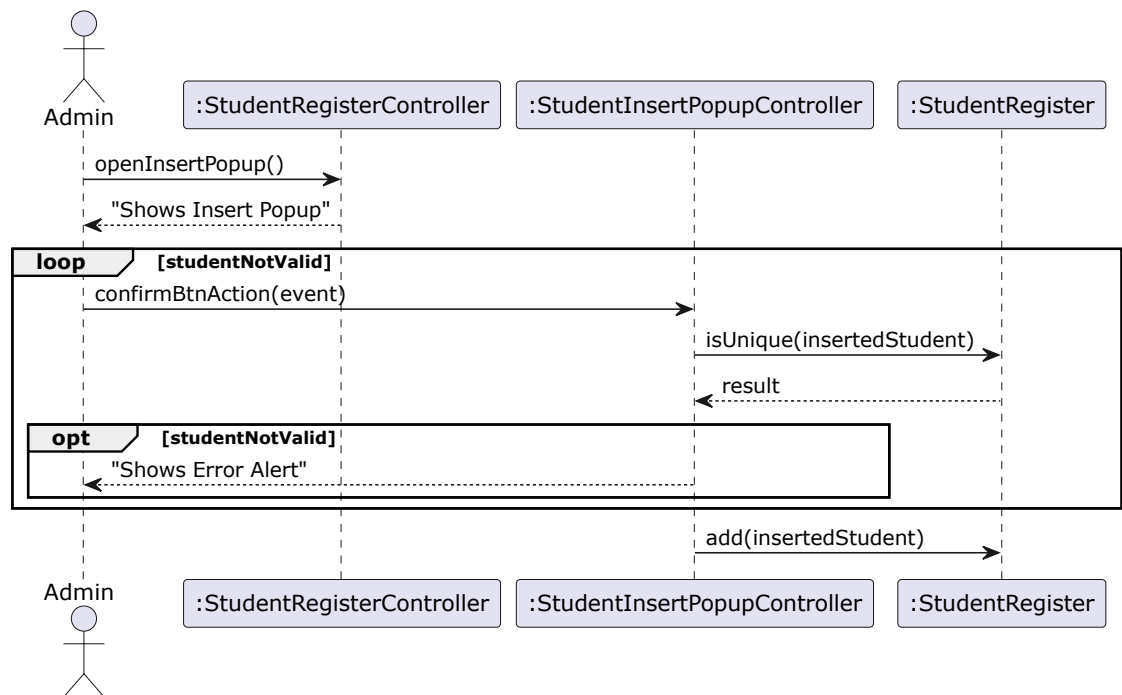


Figura 2.9: Diagramma di sequenza per la registrazione di uno studente UC-6

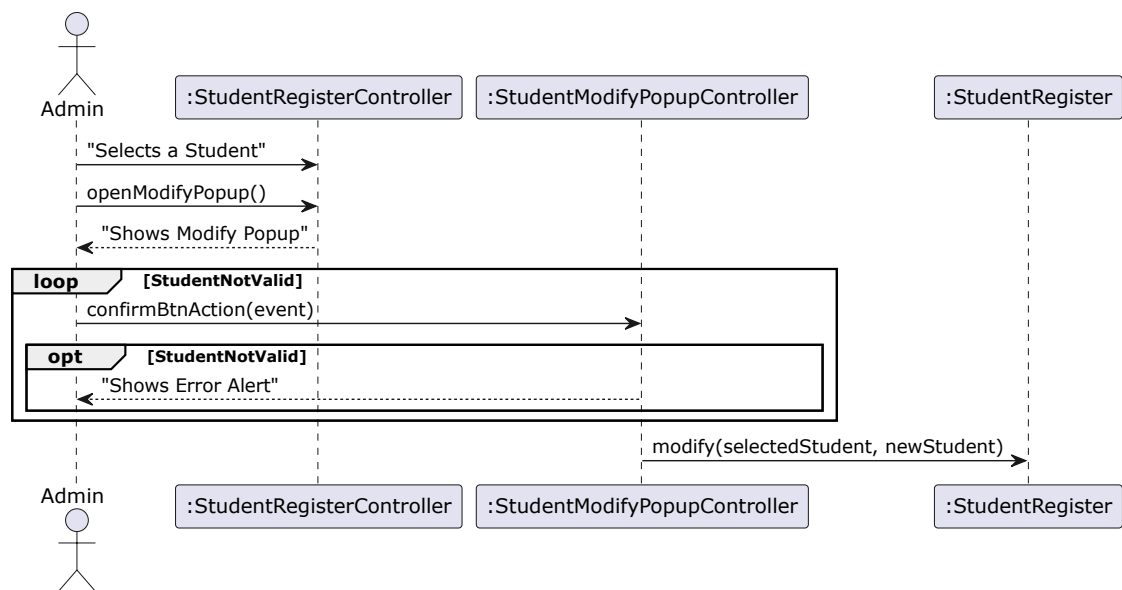


Figura 2.10: Diagramma di sequenza per la modifica di uno studente UC-7

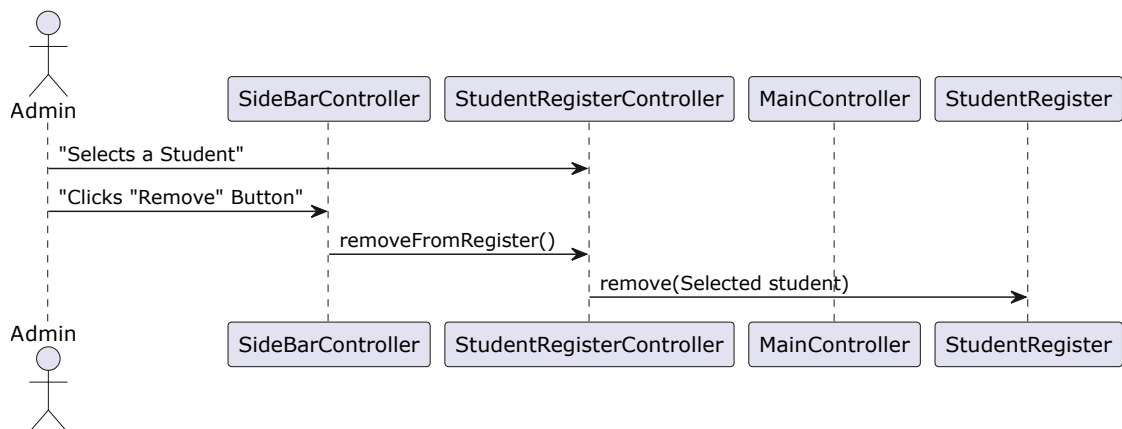


Figura 2.11: Diagramma di sequenza per la rimozione di uno studente UC-8

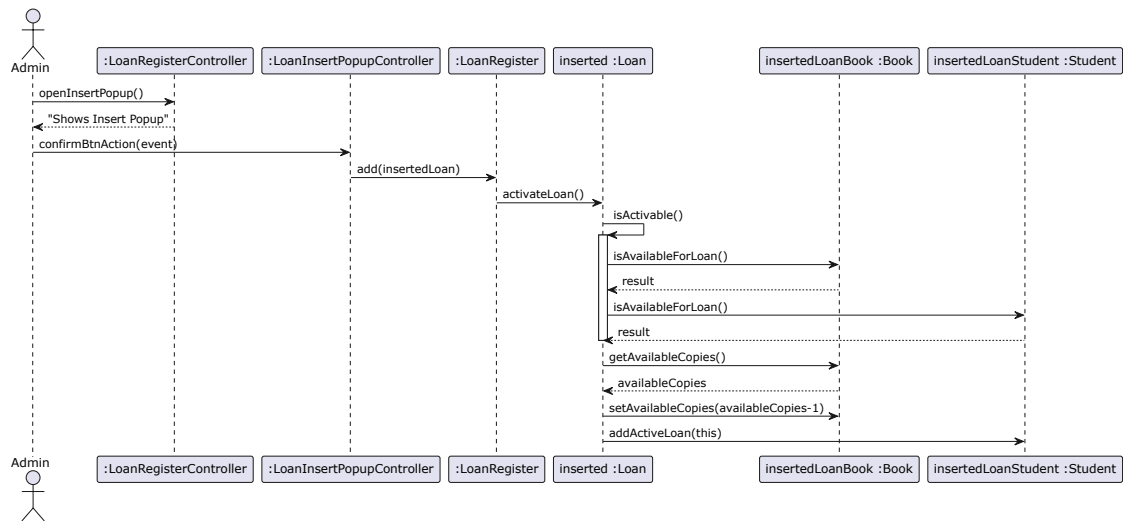


Figura 2.12: Diagramma di sequenza per la registrazione di un prestito UC-11

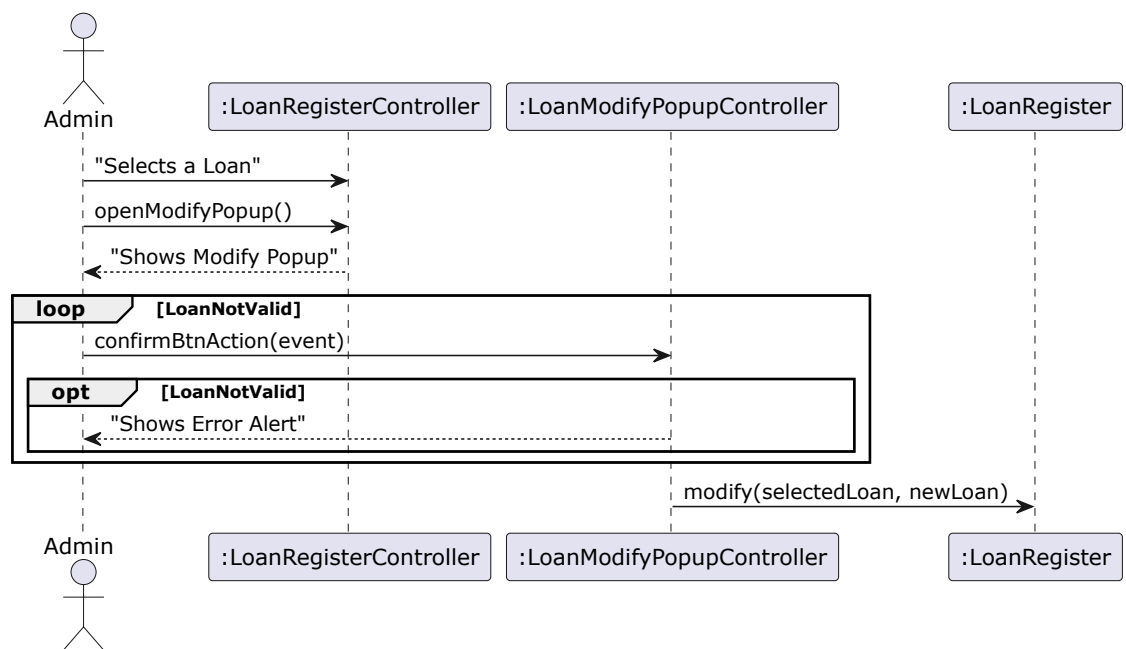


Figura 2.13: Diagramma di sequenza per la modifica di un prestito UC-12

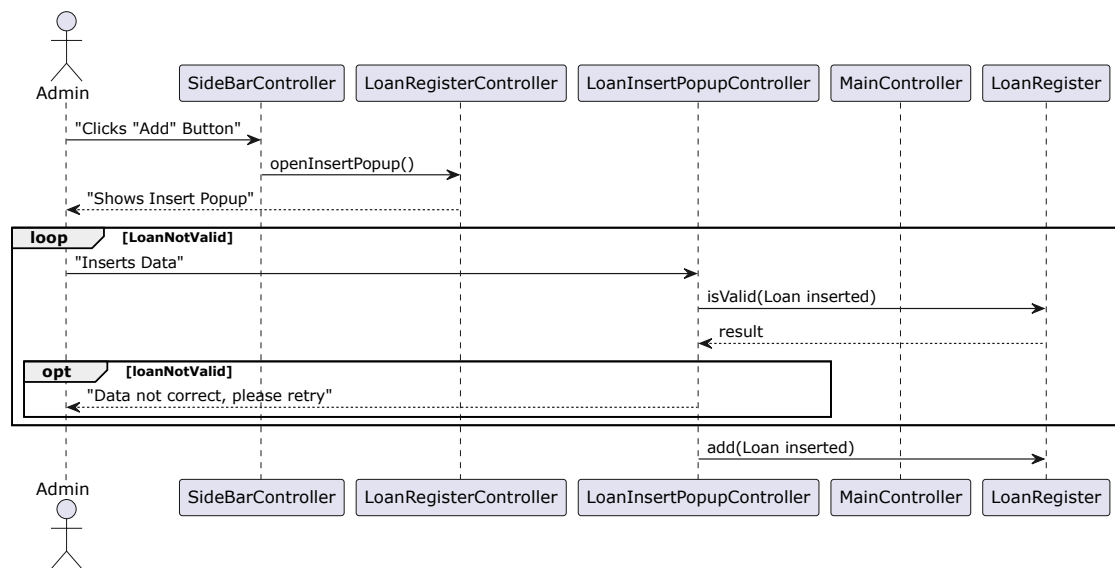


Figura 2.14: Diagramma di sequenza per la rimozione di un prestito UC-13

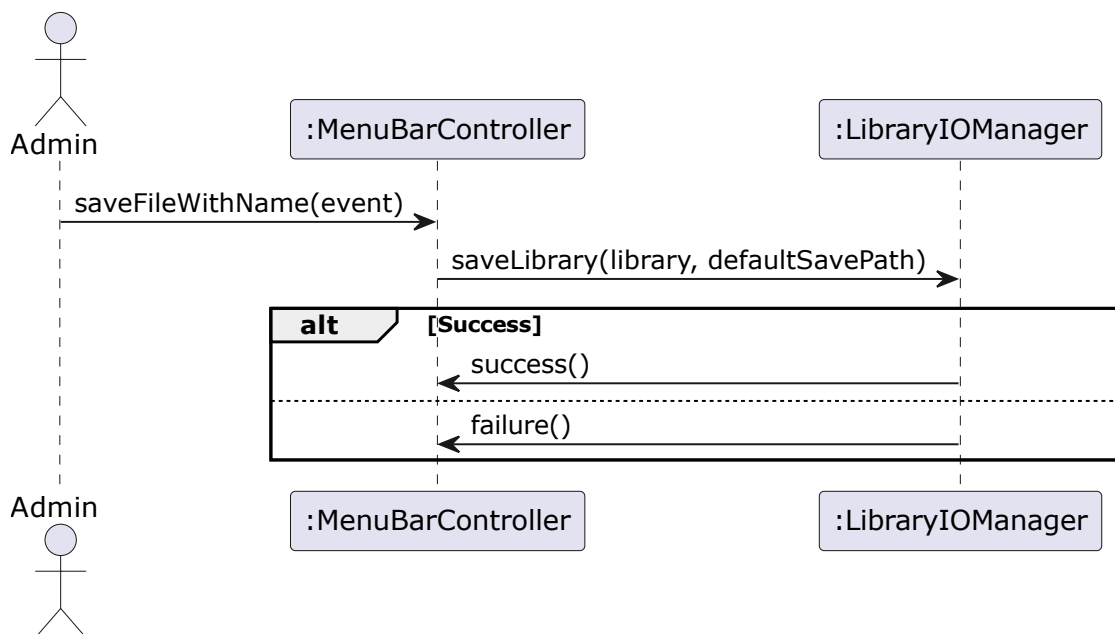


Figura 2.15: Diagramma di sequenza per il salvataggio dell'archivio (con nome) UC-17

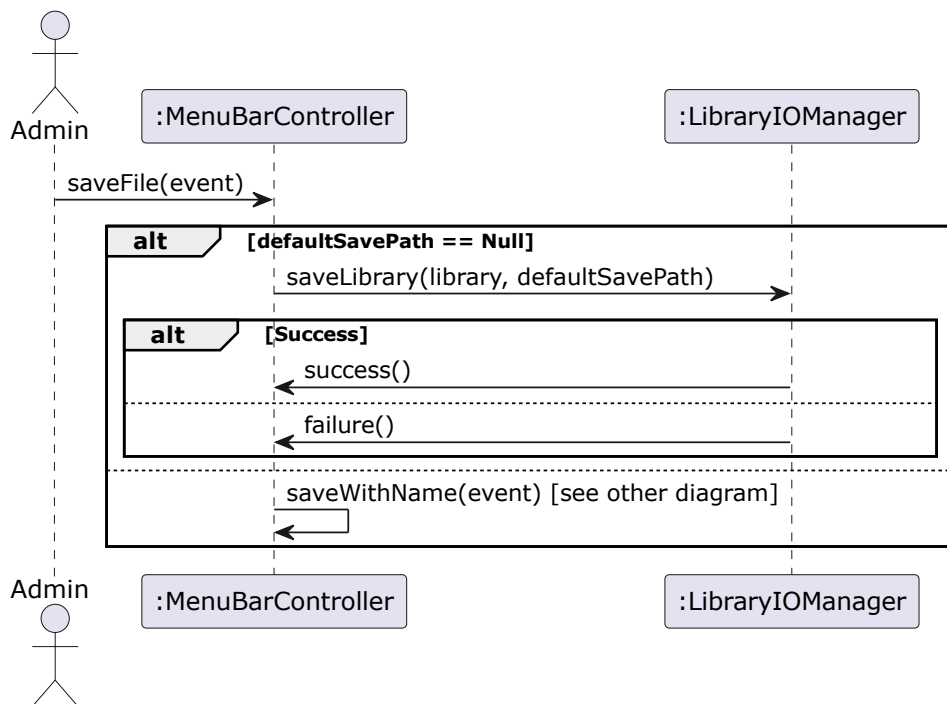


Figura 2.16: Diagramma di sequenza per il salvataggio dell'archivio UC-17

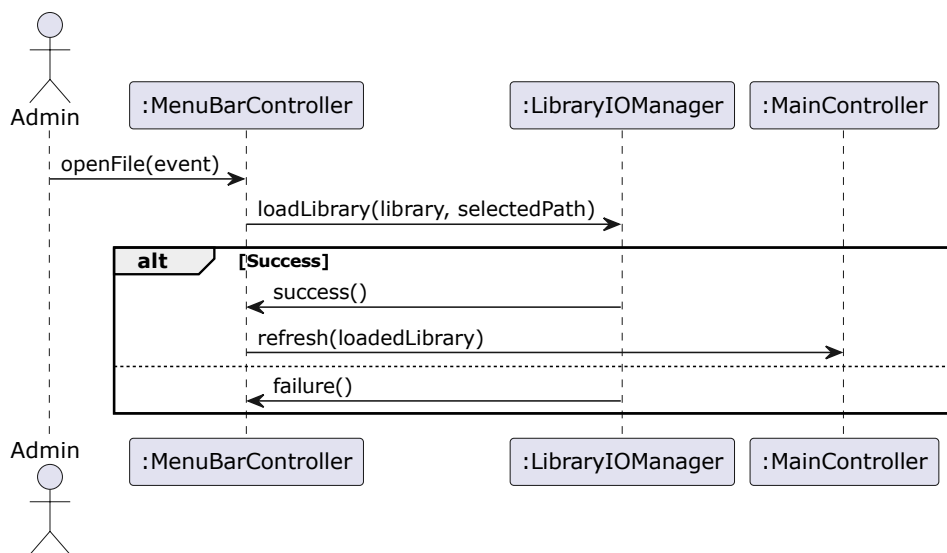


Figura 2.17: Diagramma di sequenza per il salvataggio dell'archivio UC-17

2.2.1 Livelli di coesione

Nome classe	Livello coesione	Note
Library	Funzionale	La classe aggrega tutti i register per fornire un punto di accesso unico tramite le funzioni getter.
Student, Book e Loan	Funzionale	Contiene tutti gli attributi e metodi per la visione e gestione di una singola entità.
Student, Book e Loan Register	Funzionale	Le classi implementano un registro del tipo specificato, e forniscono i metodi utili alla gestione di quest'ultimo.
LibraryIOManager	Comunicazionale	La classe implementa le funzionalità utili all'interfacciamento della biblioteca con un file esterno di memorizzazione.
MainController	Temporale	La classe implementa le funzionalità utili all'inizializzazione dei principali controller e del gestore IO.
MenuBarController	Funzionale	La classe gestisce il menu bar fornendo l'accesso alle funzione di IO tramite la classe LibraryIOManager
Student, Book e Loan RegisterController	Comunicazionale	Permette la visualizzazione dei dati in tabella, implementa le funzioni di ricerca e rimozione. Interfaccia il registro con i Popup di inserimento e modifica.
Student, Book e Loan InsertPopupController	Funzionale	La classe implementa le funzionalità utili all'interfacciamento dell'admin con il registro al fine di consentire l'inserimento di un elemento del tipo specificato.
Student, Book e Loan ModifyPopupController	Funzionale	La classe implementa le funzionalità utili all'interfacciamento dell'admin con il registro al fine di consentire la modifica di un elemento del tipo specificato.
SideBarController	Comunicazionale	La classe implementa le funzionalità utili per l'accesso alle funzionalità di gestione del registro e alla ricerca

2.2.2 Livelli di accoppiamento

Classe 1	Classe 2	Accoppiamento
Loan	Student	Per timbro
Loan	Book	Per timbro
Student	StudentRegister	Per dati
Book	BookRegister	Per dati
Loan	LoanRegister	Per dati
Library	Register	Per dati
MainController	MenuBarController	Per timbro
*RegisterController	MainController	Per dati
MenuBarController	LibraryIOManager	Per timbro
*RegisterController	SideBarController	Per dati
*RegisterController	*InsertPopupController	Per dati
*RegisterController	*ModifyPopupController	Per dati

L'accoppiamento per timbro tra **Loan** e **Student/Book** sussiste in quanto al **Loan** non sono necessarie tutte le funzionalità fornite dalle due classi. Sebbene fosse possibile ridurre l'accoppiamento mediante introduzione di opportune interfacce ciò avrebbe complicato la stampa del registro prestiti. Si è dunque ritenuto questo accoppiamento accettabile.

L'accoppiamento per timbro tra **MainController** e **MenuBarController** sussiste in quando il medesimo riceve dal **MainController** una **Library** che gli dà accesso a più funzionalità di quante ne sono effettivamente richieste. Per le stesse motivazioni sussiste un accoppiamento per timbro tra **MenuBarController** e **LibraryIOManager** (la struttura dati passata è la stessa).