

Relazioni UML del Sistema Biblioteca Universitaria

Contents

1 Relazioni tra le classi del modello UML	2
1.1 Diagrammi UML di riferimento	2
1.2 Descrizione dei package	5
1.2.1 Library	5
1.2.2 Package User	5
1.2.3 Package Loan	6
1.2.4 Package Book	6
1.2.5 Package Author	7
1.3 Relazioni organizzate per package	8
1.3.1 Library	8
1.3.2 Package User	8
1.3.3 Package Loan	9
1.3.4 Package Book	9
1.3.5 Package Author	10
2 Diagrammi di Sequenza	11
2.1 UC-1 — Visualizza lista libri	12
2.2 UC-2 — Cerca libro	13
2.3 UC-3 — Modifica libro	14
2.4 UC-4 — Rimuovi libro	15
2.5 UC-5 — Aggiungi libro	16
2.6 UC-6 — Visualizza lista prestiti	17
2.7 UC-7 — Cerca prestito	18
2.8 UC-8 — Registra prestito	19
2.9 UC-9 — Registra restituzione	20
2.10 UC-10 — Salva archivio	21
2.11 UC-11 — Visualizza lista utenti	22
2.12 UC-12 — Cerca utente	23
2.13 UC-13 — Modifica utente	24
2.14 UC-14 — Elimina utente	25
2.15 UC-15 — Inserisci utente	26

Chapter 1

Relazioni tra le classi del modello UML

1.1 Diagrammi UML di riferimento

Di seguito sono riportati i due diagrammi delle classi che costituiscono l'intero modello UML del sistema Biblioteca Universitaria. Essi rappresentano la struttura dei package principali, le classi coinvolte, i relativi metodi e le relazioni tra esse.

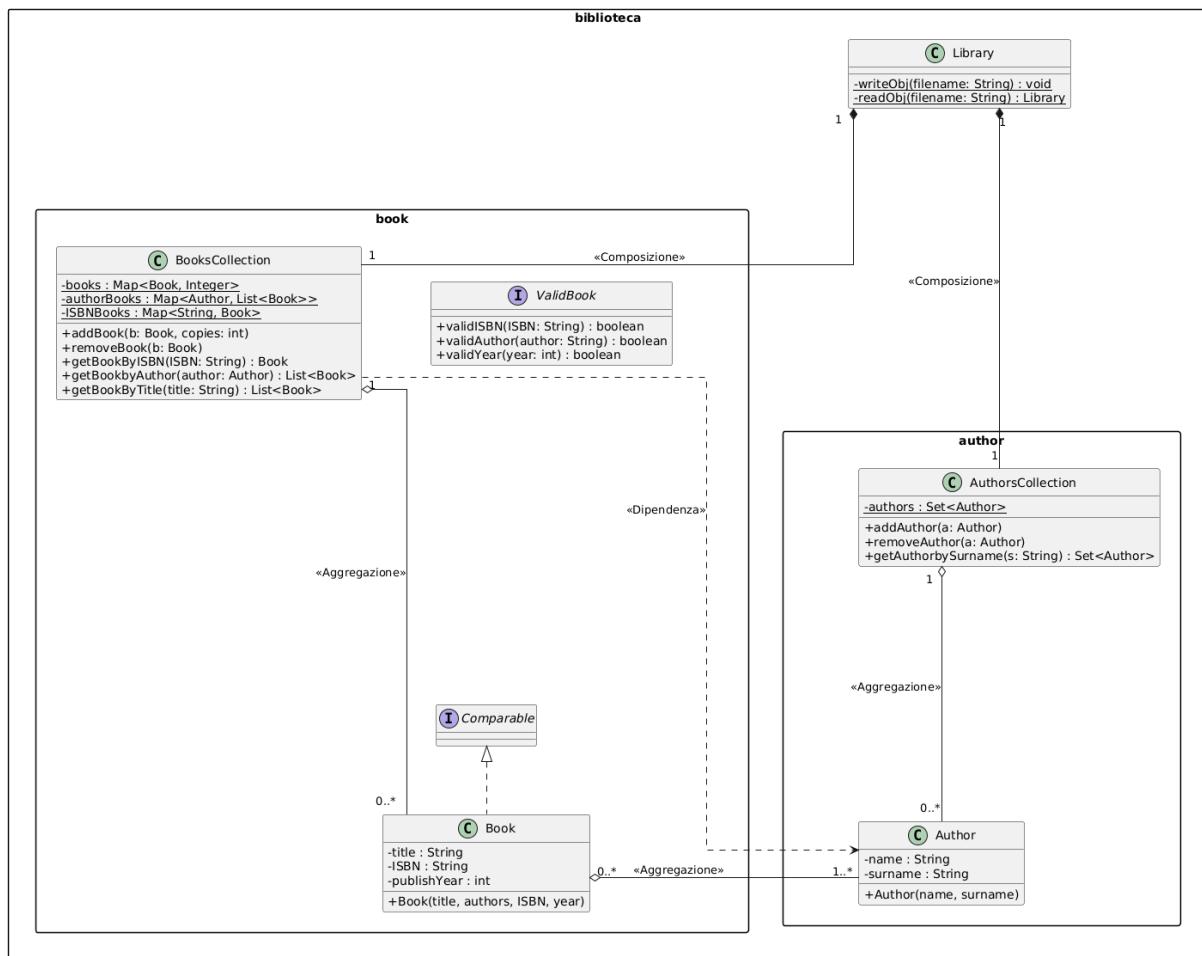


Figure 1.1: Diagramma UML — Package Book e Author

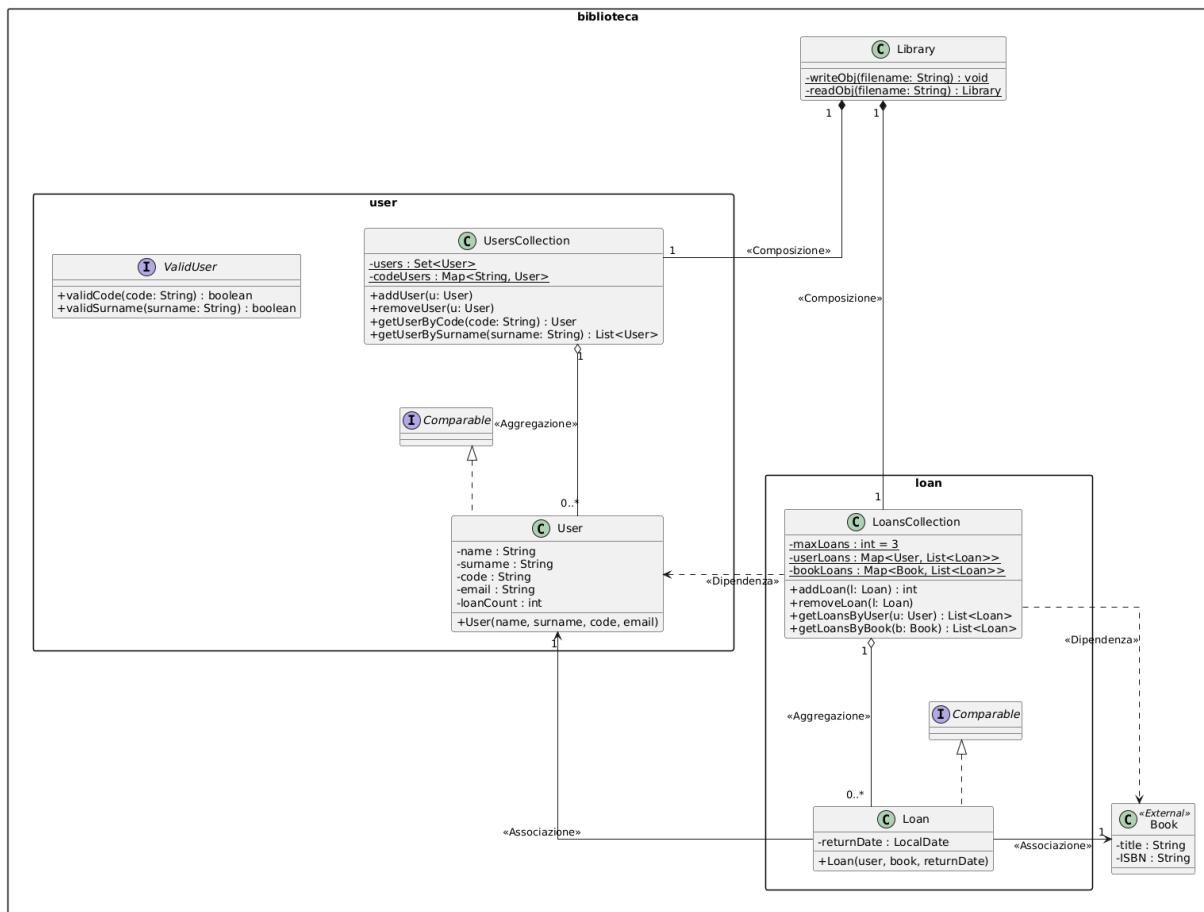


Figure 1.3: Diagramma UML — Package User e Loan

1.2 Descrizione dei package

1.2.1 Library

Library ha il ruolo di gestire l'intero sistema. La classe **Library** gestisce il salvataggio e il caricamento dell'intero archivio attraverso i metodi:

- `- writeObj(filename : String) : void` — serializza e salva l'intero archivio su file.
- `- readObj(filename : String) : Library` — legge da file e ricostruisce gli oggetti.

Essa possiede relazioni di composizione con:

- **UsersCollection**
- **LoansCollection**
- **BooksCollection**
- **AuthorsCollection**

1.2.2 Package User

Il package **User** si occupa della gestione degli utenti registrati e della memorizzazione delle loro informazioni personali. Le classi principali sono:

User:

- Attributi: `name, surname, code, email, loanCount`
- Costruttore: `User(name, surname, code, email)`

UsersCollection:

- Attributi: `users : Set<User>, codeUsers : Map<String, User>`
- Metodi:
 - `+ addUser(u : User)`
 - `+ removeUser(u : User)`
 - `+ getUserByCode(code : String) : User`
 - `+ getUserBySurname(surname : String) : List<User>`

1.2.3 Package Loan

Il package **Loan** gestisce i prestiti dei libri agli utenti e assicura il rispetto dei vincoli di sistema (massimo numero di prestiti, restituzioni, ecc.).

Loan:

- Attributo: `returnDate : LocalDate`
- Costruttore: `Loan(user, book, returnDate)`

LoansCollection:

- Attributi:
 - `maxLoans : int = 3`
 - `userLoans : Map<User, List<Loan>`
 - `bookLoans : Map<Book, List<Loan>`
- Metodi:
 - `+ addLoan(l : Loan) : int`
 - `+ removeLoan(l : Loan)`
 - `+ getLoansByUser(u : User) : List<Loan>`
 - `+ getLoansByBook(b : Book) : List<Loan>`

1.2.4 Package Book

Il package **Book** si occupa della gestione del catalogo dei libri.

Book:

- Attributi: `title, ISBN, publishYear`
- Costruttore: `Book(title, authors, ISBN, year)`

BooksCollection:

- Attributi:
 - `books : Map<Book, Integer>`
 - `authorBooks : Map<Author, List<Book>`
 - `ISBNBooks : Map<String, Book>`
- Metodi:
 - `+ addBook(b : Book, copies : int)`

- + removeBook(b : Book)
- + getBookByISBN(ISBN : String) : Book
- + getBookByAuthor(author : Author) : List<Book>
- + getBookByTitle(title : String) : List<Book>

1.2.5 Package Author

Il package **Author** si occupa della gestione delle informazioni relative agli autori.

Author:

- Attributi: name, surname
- Costruttore: Author(name, surname)

AuthorsCollection:

- Attributo: authors : Set<Author>
- Metodi:
 - + addAuthor(a : Author)
 - + removeAuthor(a : Author)
 - + getAuthorBySurname(s : String) : Set<Author>

1.3 Relazioni organizzate per package

1.3.1 Library

Library → UsersCollection (Composizione)

La relazione di composizione indica che la collezione degli utenti è parte integrante dell'oggetto Library.

Coesione: Funzionale

Accoppiamento: Necessario

Library → LoansCollection (Composizione)

Come sopra, ma per i prestiti.

Coesione: Funzionale

Accoppiamento: Necessario

Library → BooksCollection (Composizione)

La gestione del catalogo appartiene alla libreria.

Coesione: Funzionale

Accoppiamento: Necessario

Library → AuthorsCollection (Composizione)

La collezione degli autori è parte integrante del sistema.

Coesione: Funzionale

Accoppiamento: Necessario

1.3.2 Package User

UsersCollection → User (Aggregazione)

UsersCollection contiene utenti autonomi.

Coesione: Comunicazionale

Accoppiamento: Per dati

Loan → User (Associazione)

Ogni prestito è associato ad un utente.

Coesione: Funzionale

Accoppiamento: Per dati

LoansCollection → User (Dipendenza)

LoansCollection utilizza gli utenti nelle sue mappe.

Coesione: Sequenziale

Accoppiamento: Per dati

1.3.3 Package Loan

LoansCollection → Loan (Aggregazione)

La collezione contiene prestiti autonomi.

Coesione: Comunicazionale

Accoppiamento: Per dati

Loan → Book (Associazione)

Un prestito fa riferimento ad un libro.

Coesione: Funzionale

Accoppiamento: Per dati

LoansCollection → Book (Dipendenza)

LoansCollection utilizza Book per le mappe interne.

Coesione: Sequenziale

Accoppiamento: Per dati

1.3.4 Package Book

BooksCollection → Book (Aggregazione)

La collezione contiene libri autonomi.

Coesione: Comunicazionale

Accoppiamento: Per dati

Book → Author (Aggregazione)

Un libro è associato a più autori.

Coesione: Comunicazionale

Accoppiamento: Per dati

BooksCollection → Author (Dipendenza)

BooksCollection usa Author per ricerche e indicizzazioni.

Coesione: Logica

Accoppiamento: Per dati

1.3.5 Package Author

AuthorsCollection → Author (Aggregazione)

La collezione contiene autori autonomi.

Coesione: Comunicazionale

Accoppiamento: Per dati

Chapter 2

Diagrammi di Sequenza

In questo capitolo sono riportati tutti i diagrammi di sequenza relativi ai casi d'uso principali del sistema Biblioteca Universitaria. Ogni diagramma descrive il flusso operativo tra attori e componenti dell'applicazione per l'esecuzione di una specifica funzionalità.

2.1 UC-1 — Visualizza lista libri

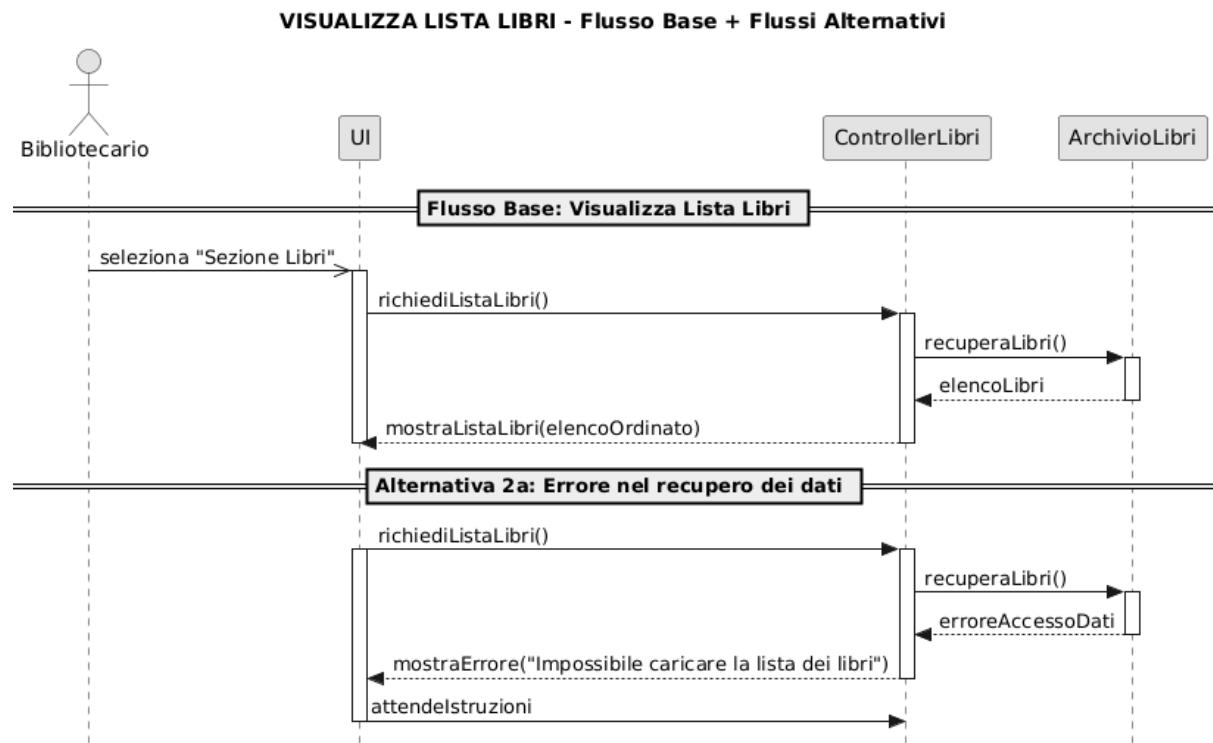


Figure 2.1: Diagramma di Sequenza — UC-1 Visualizza lista libri

2.2 UC-2 — Cerca libro

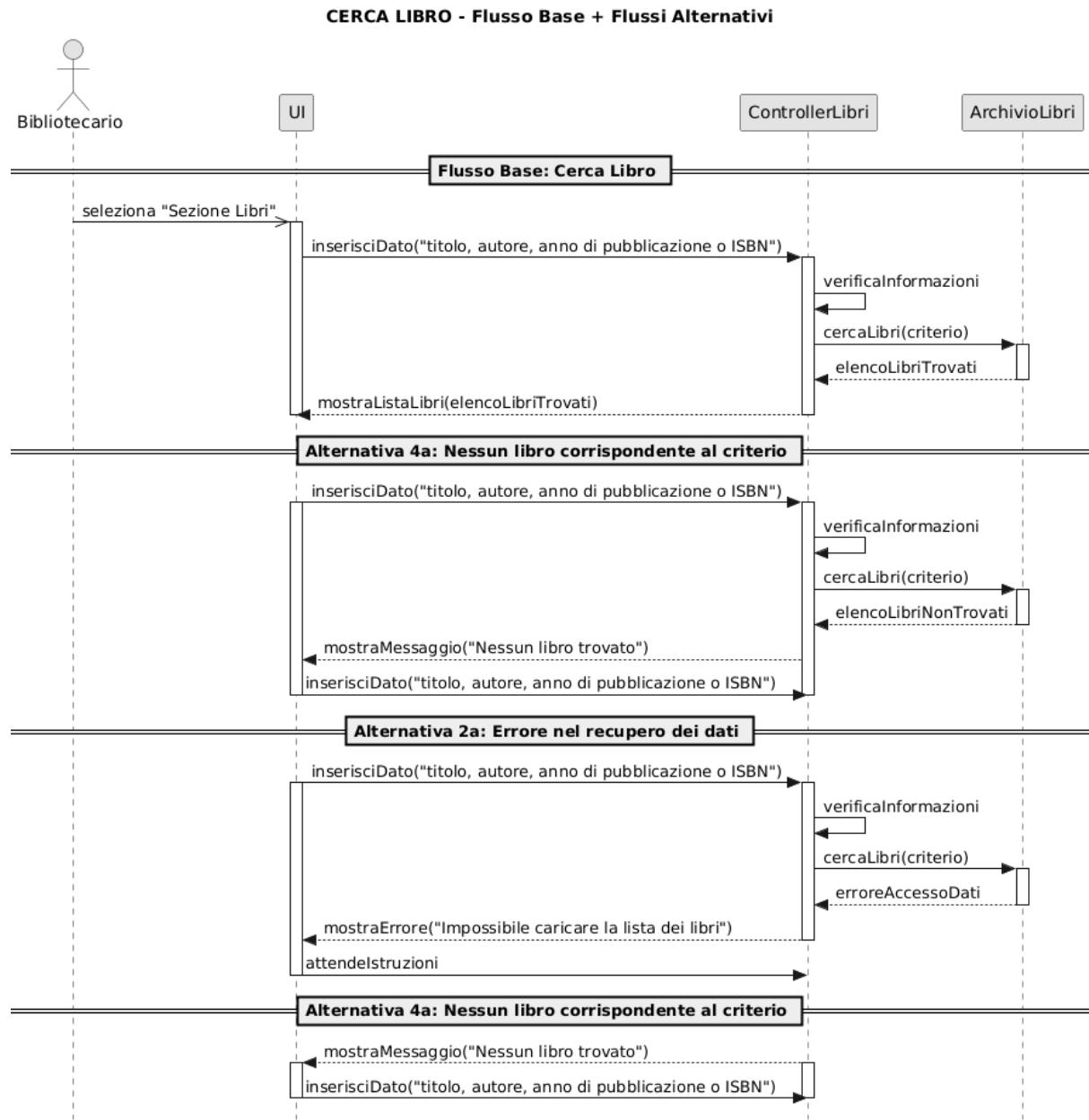


Figure 2.2: Diagramma di Sequenza — UC-2 Cerca libro

2.3 UC-3 — Modifica libro

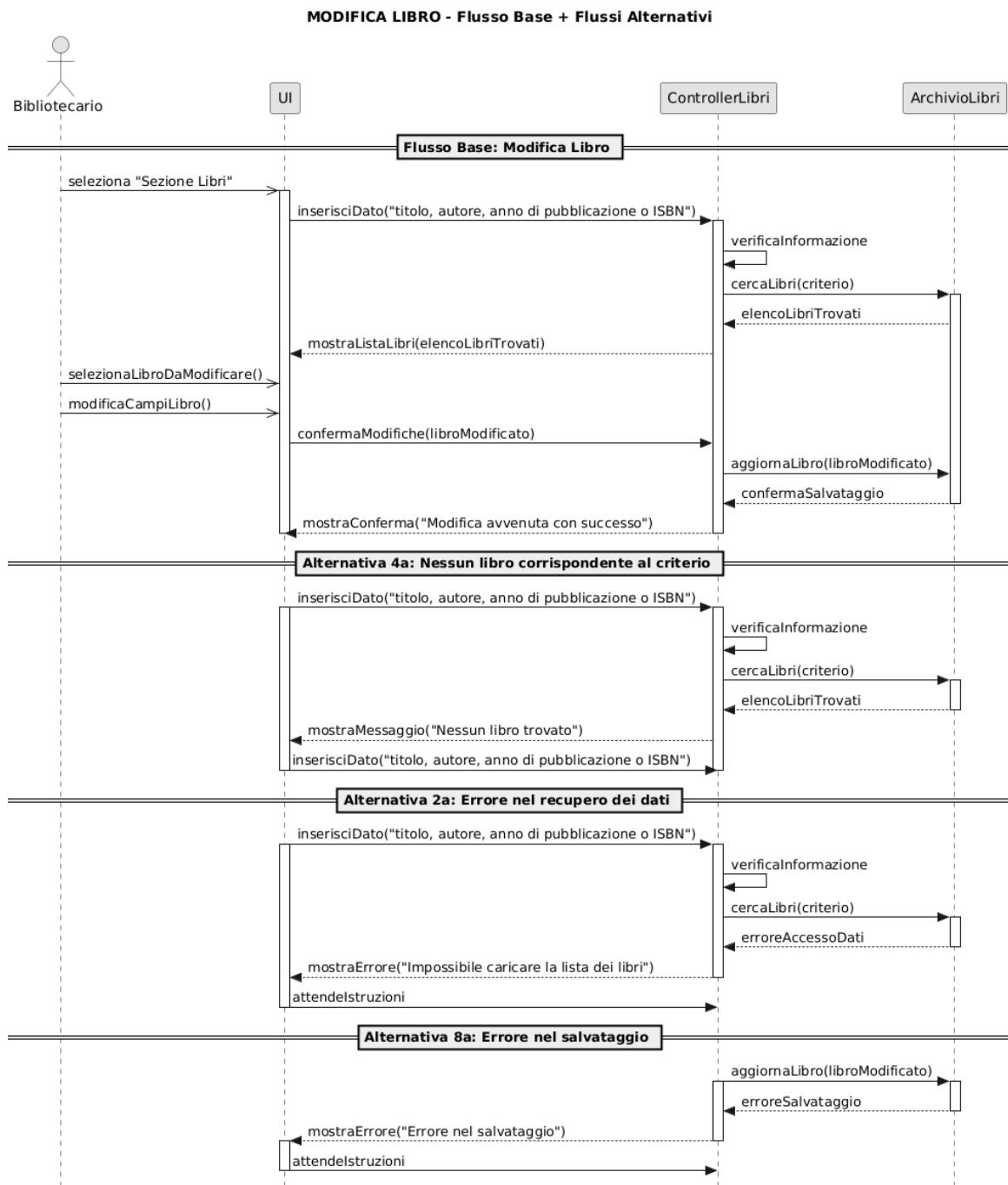


Figure 2.3: Diagramma di Sequenza — UC-3 Modifica libro

2.4 UC-4 — Rimuovi libro

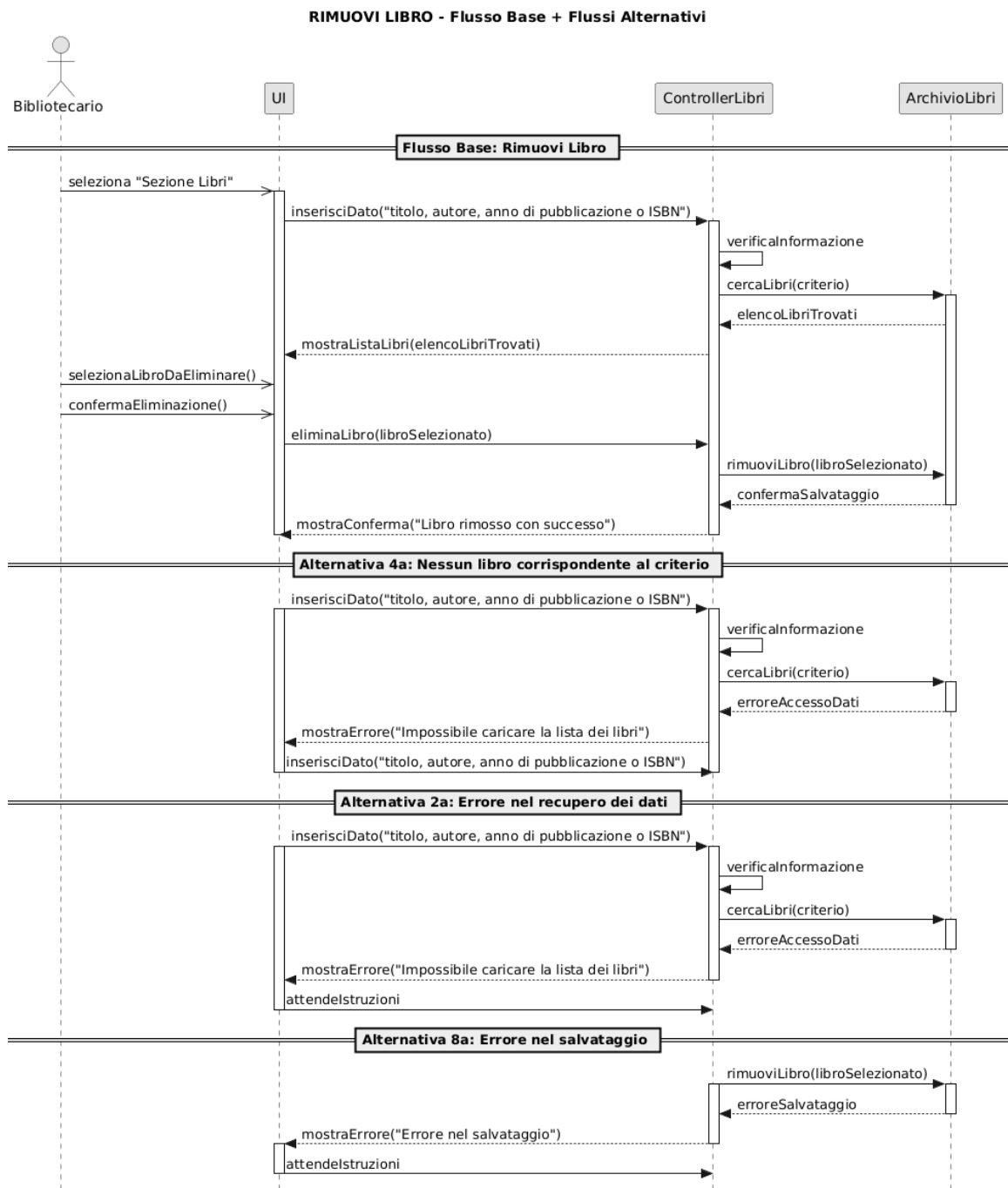


Figure 2.4: Diagramma di Sequenza — UC-4 Rimuovi libro

2.5 UC-5 — Aggiungi libro

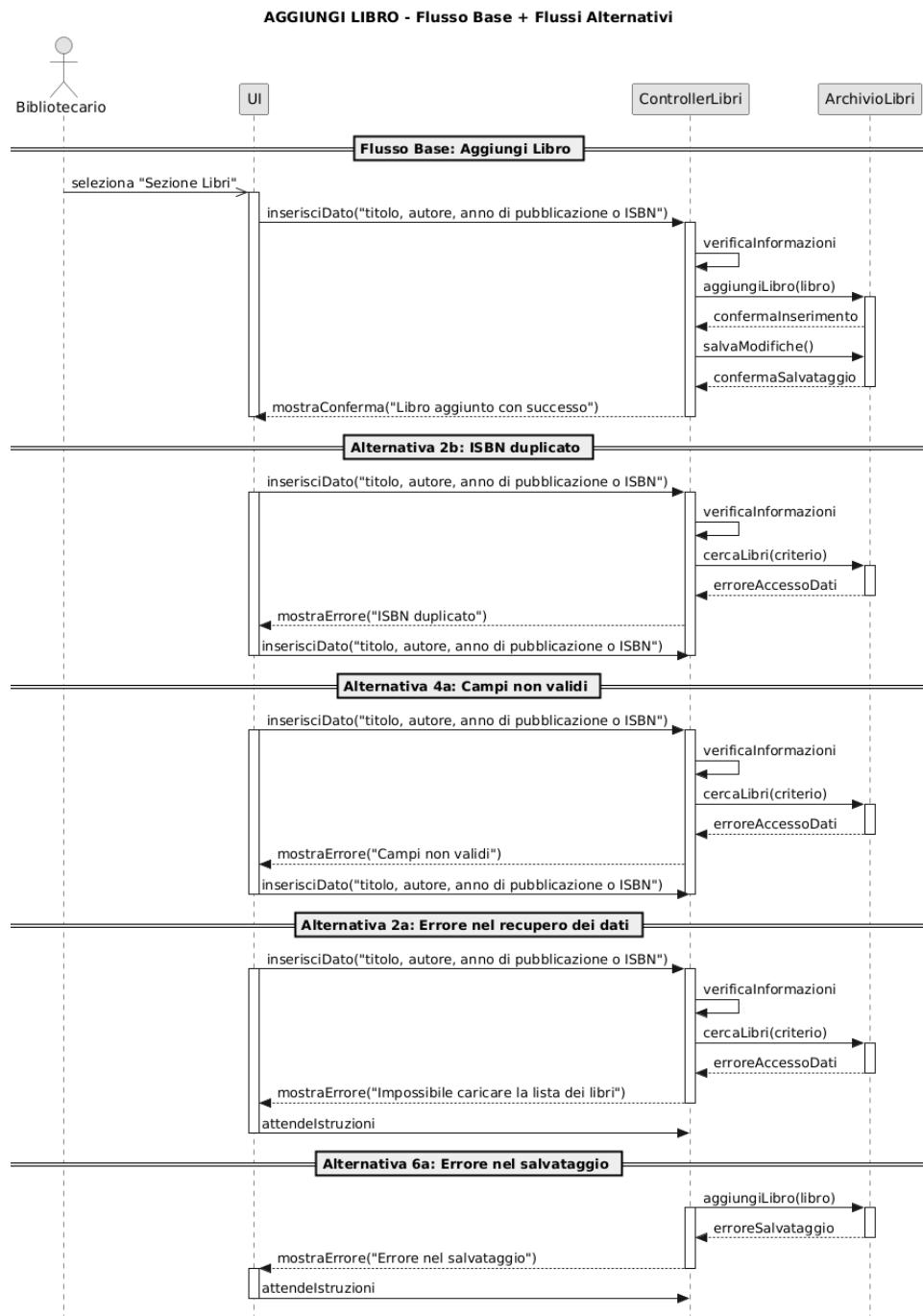


Figure 2.5: Diagramma di Sequenza — UC-5 Aggiungi libro

2.6 UC-6 — Visualizza lista prestiti

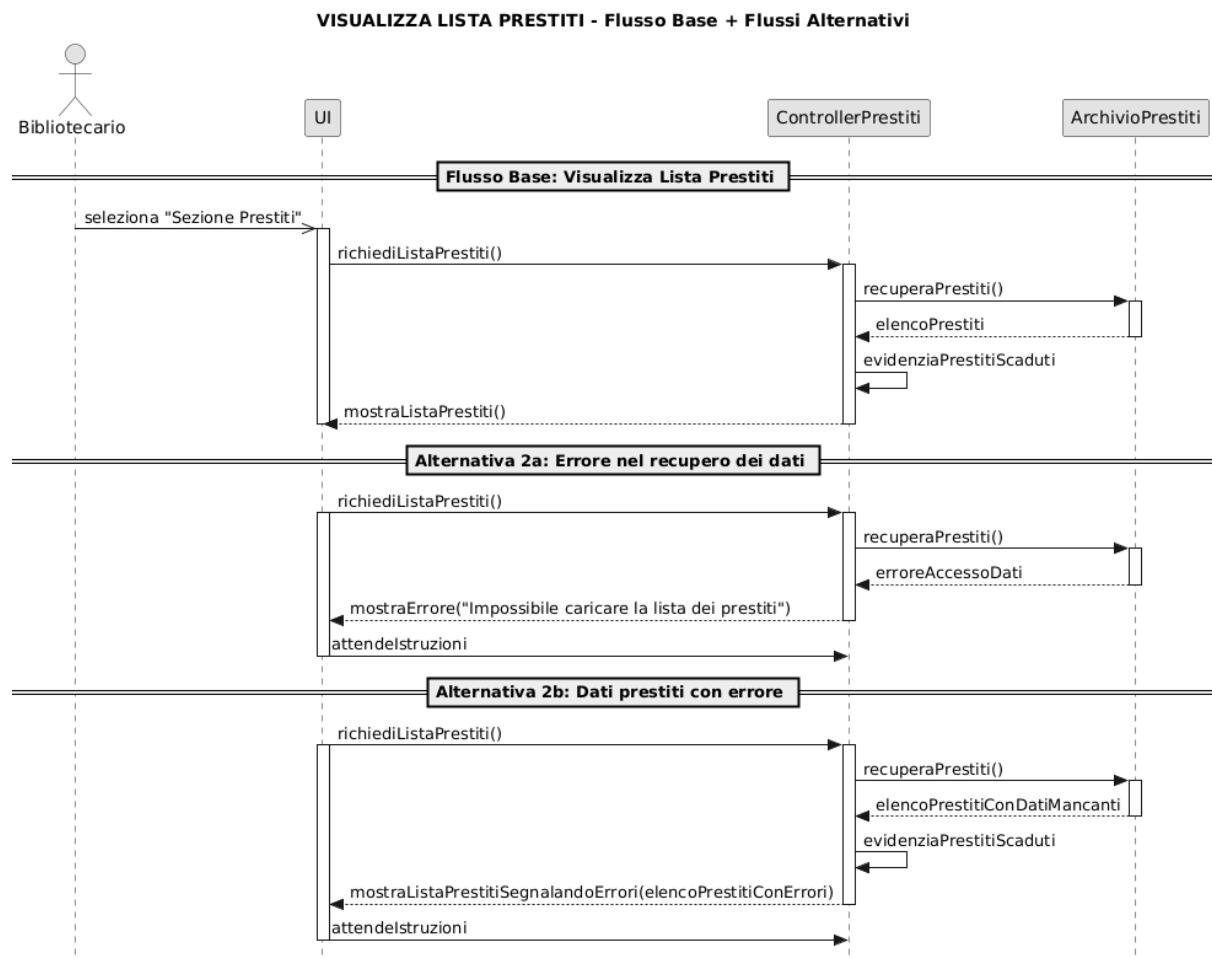


Figure 2.6: Diagramma di Sequenza — UC-6 Visualizza lista prestiti

2.7 UC-7 — Cerca prestito

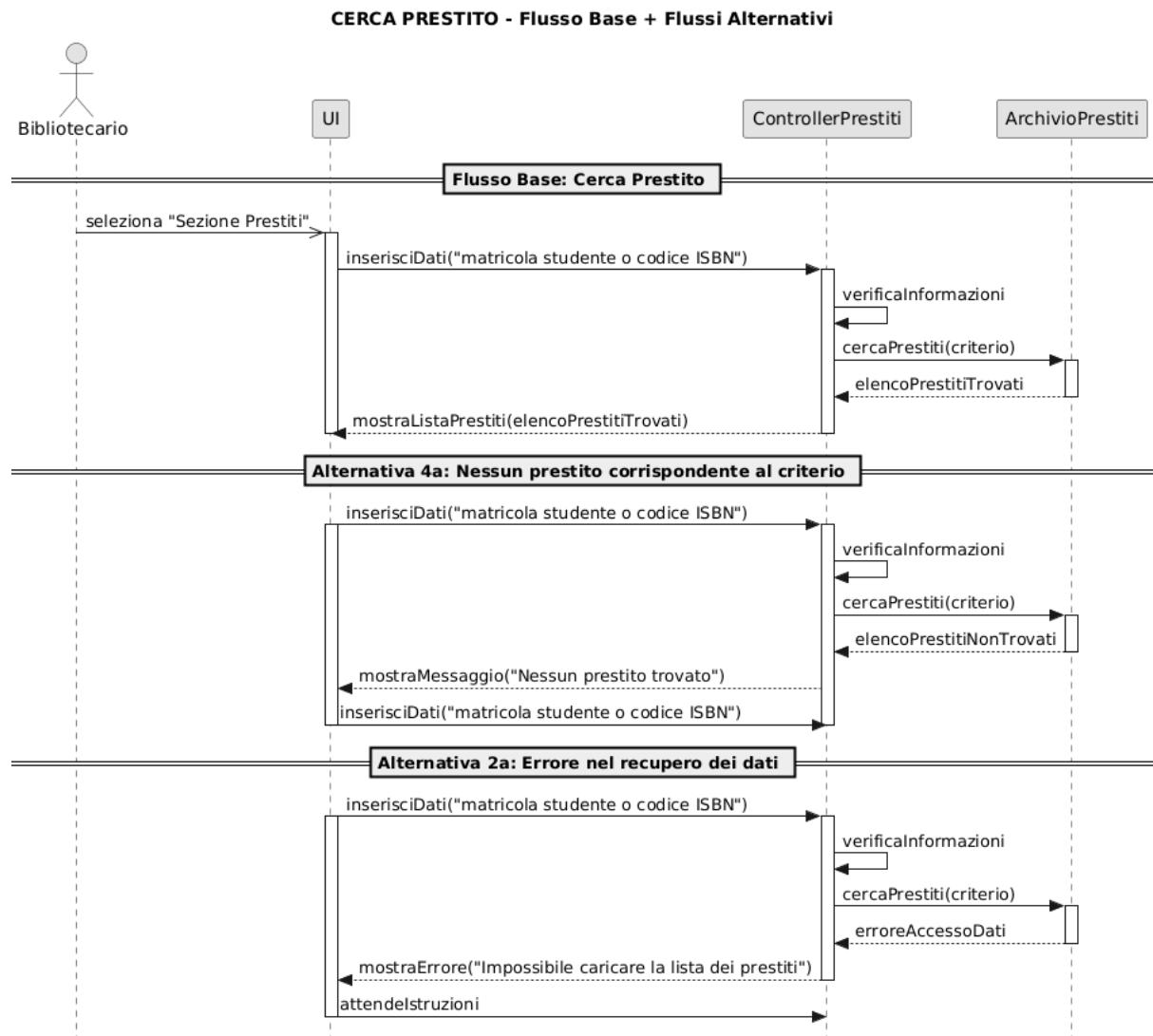


Figure 2.7: Diagramma di Sequenza — UC-7 Cerca prestito

2.8 UC-8 — Registra prestito

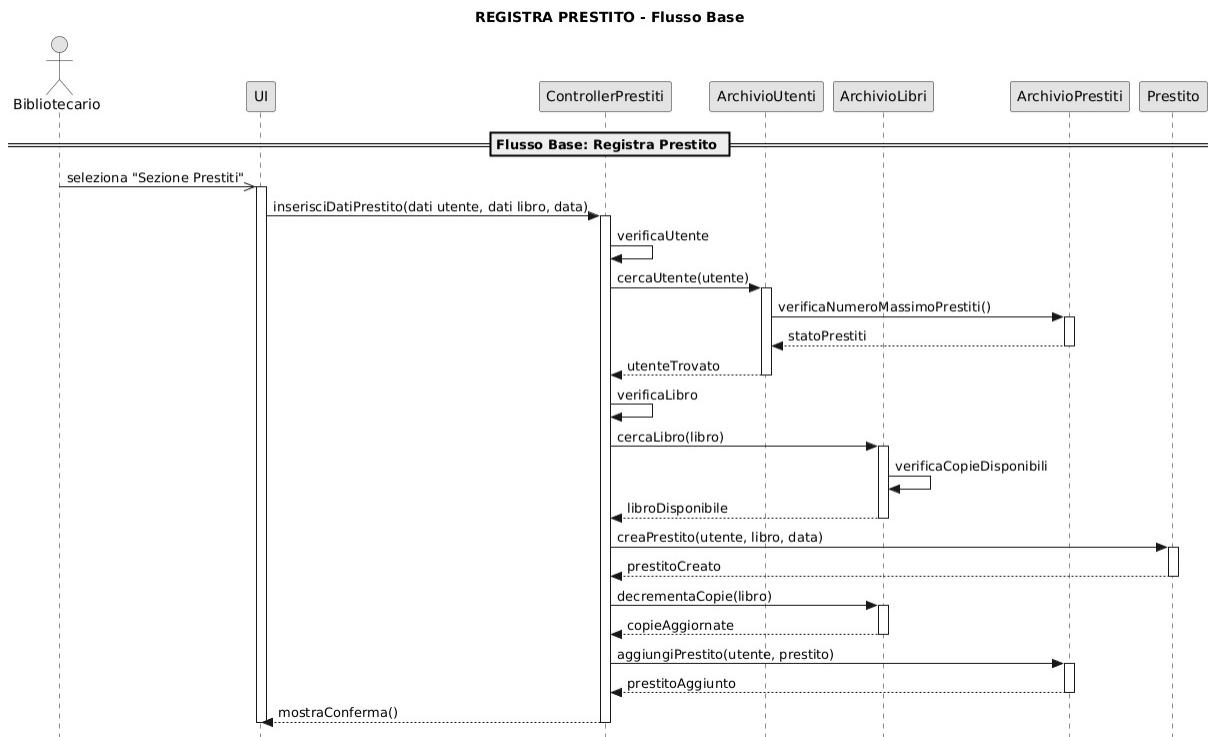


Figure 2.8: Diagramma di Sequenza — UC-8 Registra prestito

2.9 UC-9 — Registra restituzione

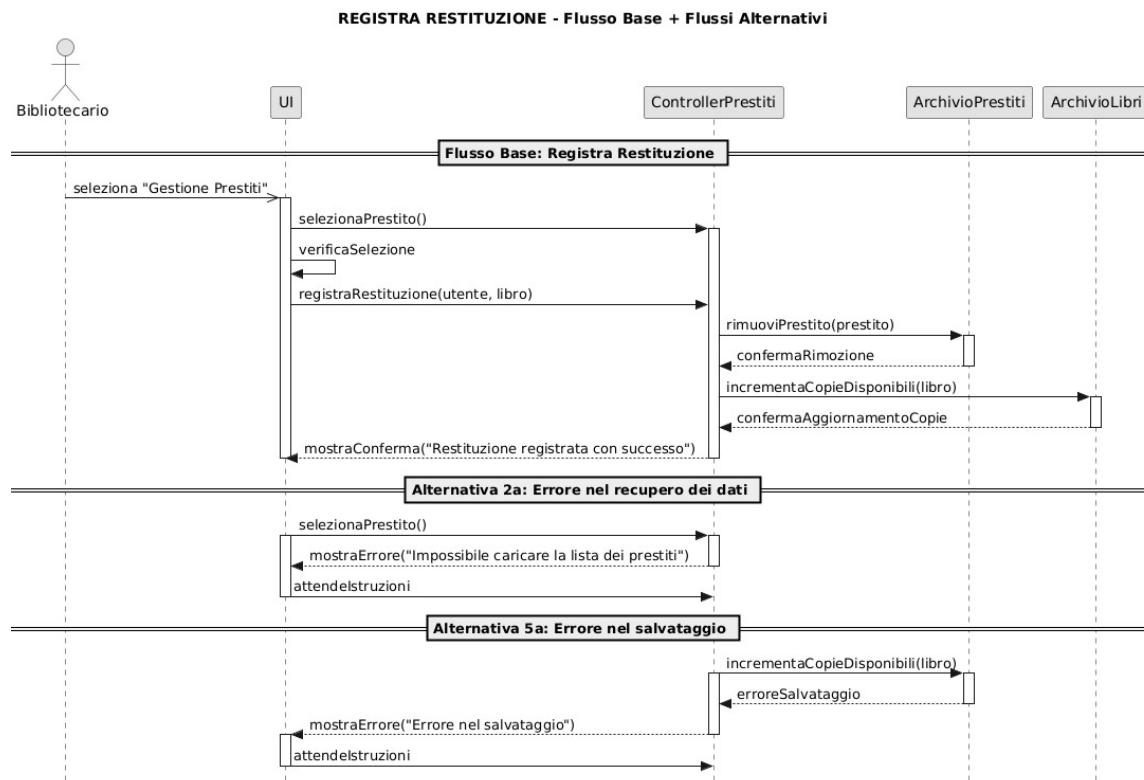


Figure 2.9: Diagramma di Sequenza — UC-9 Registra restituzione

2.10 UC-10 — Salva archivio

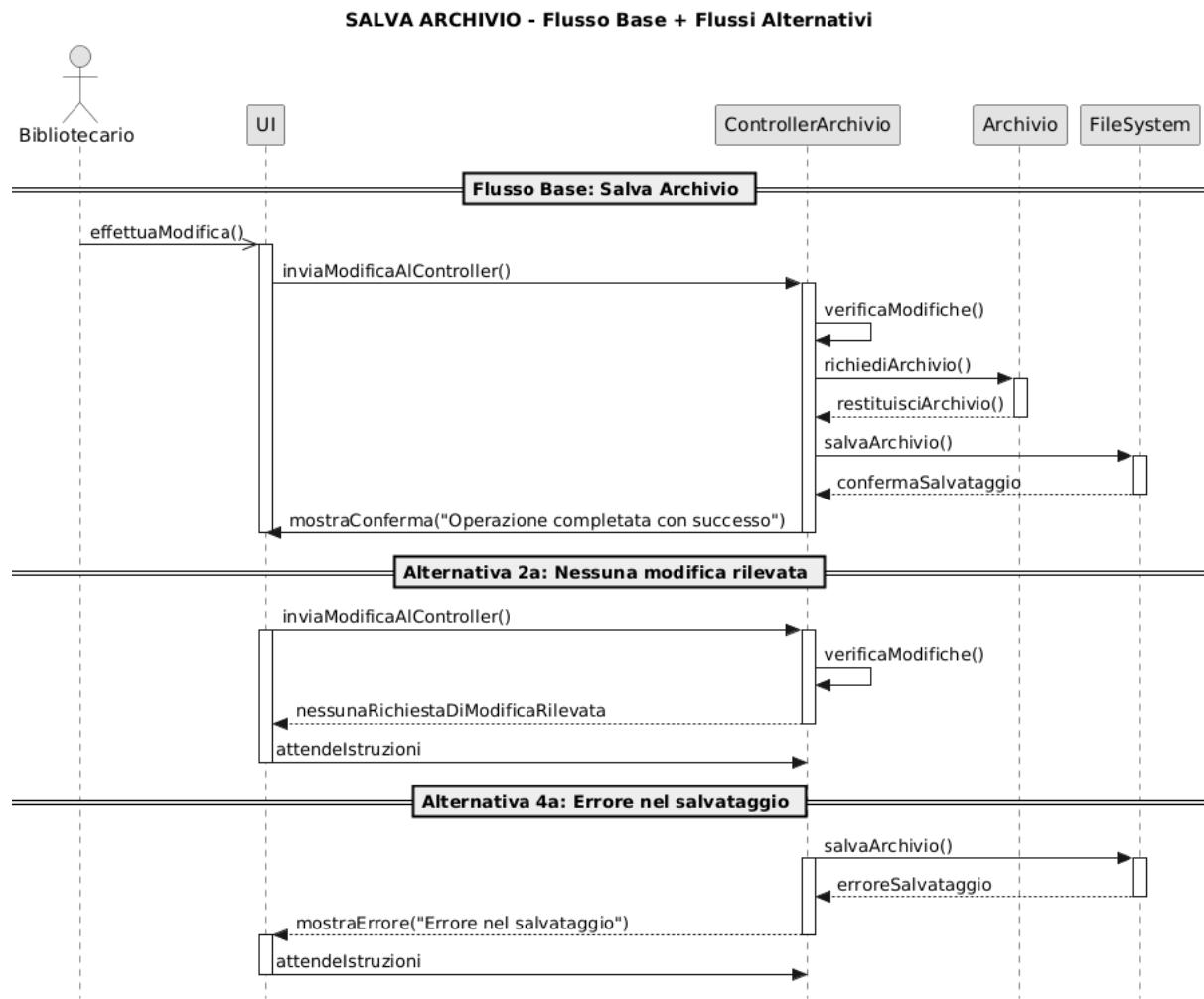


Figure 2.10: Diagramma di Sequenza — UC-10 Salva archivio

2.11 UC-11 — Visualizza lista utenti

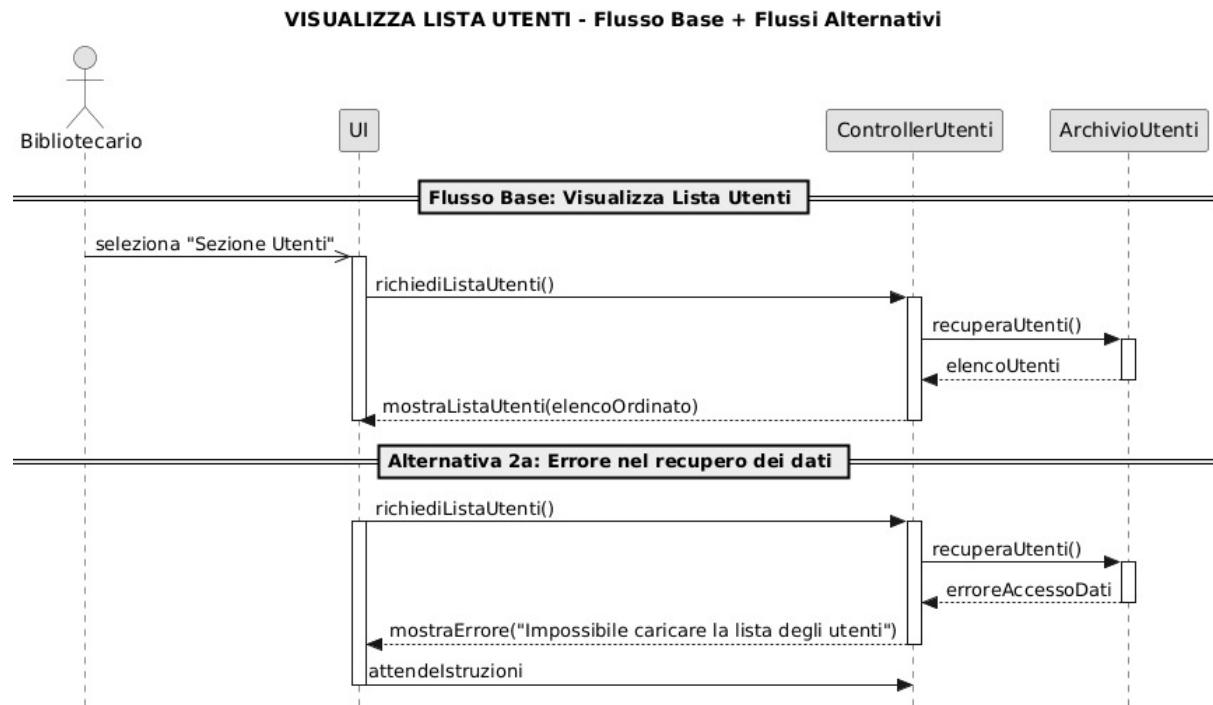


Figure 2.11: Diagramma di Sequenza — UC-11 Visualizza lista utenti

2.12 UC-12 — Cerca utente

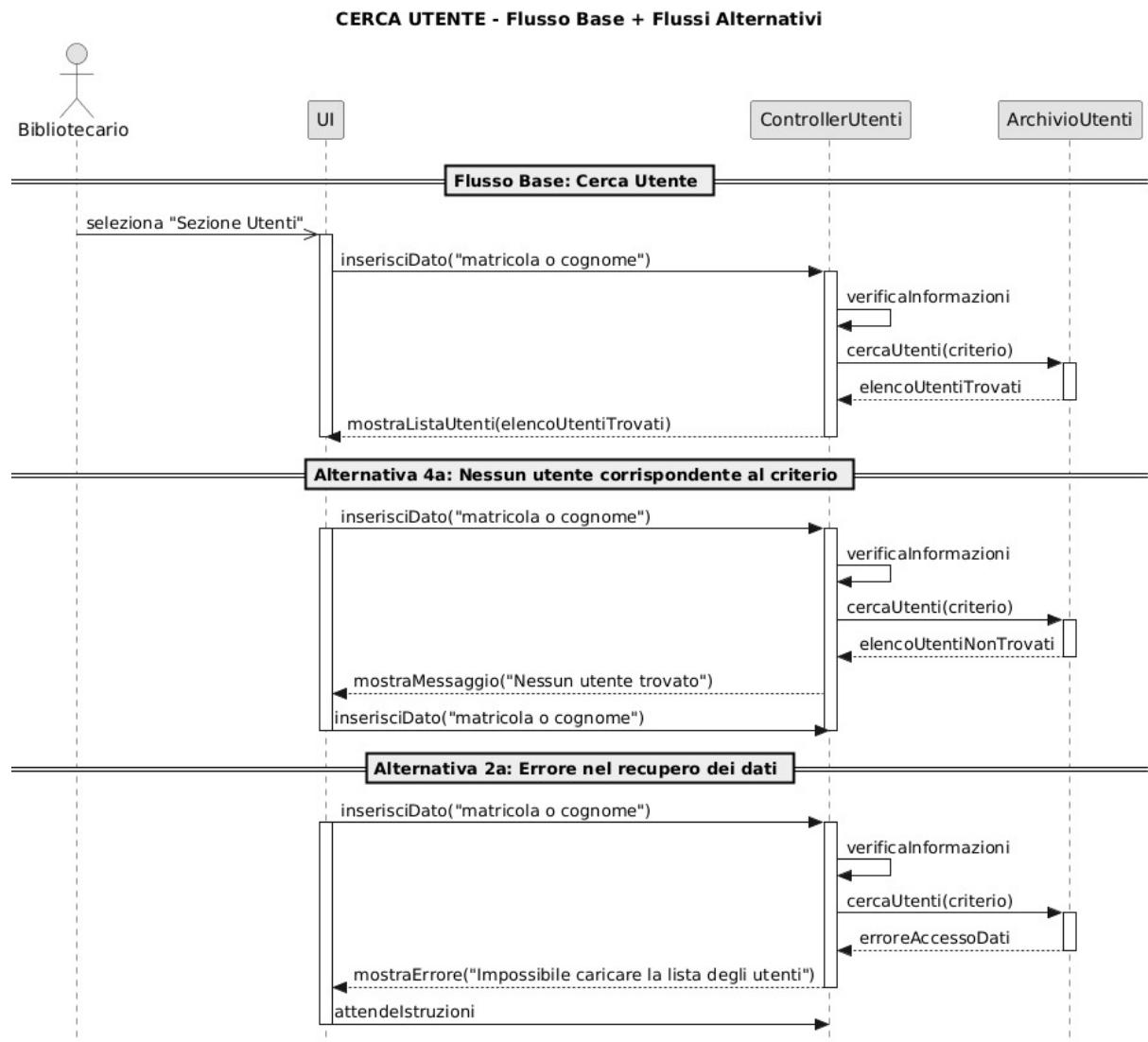


Figure 2.12: Diagramma di Sequenza — UC-12 Cerca utente

2.13 UC-13 — Modifica utente

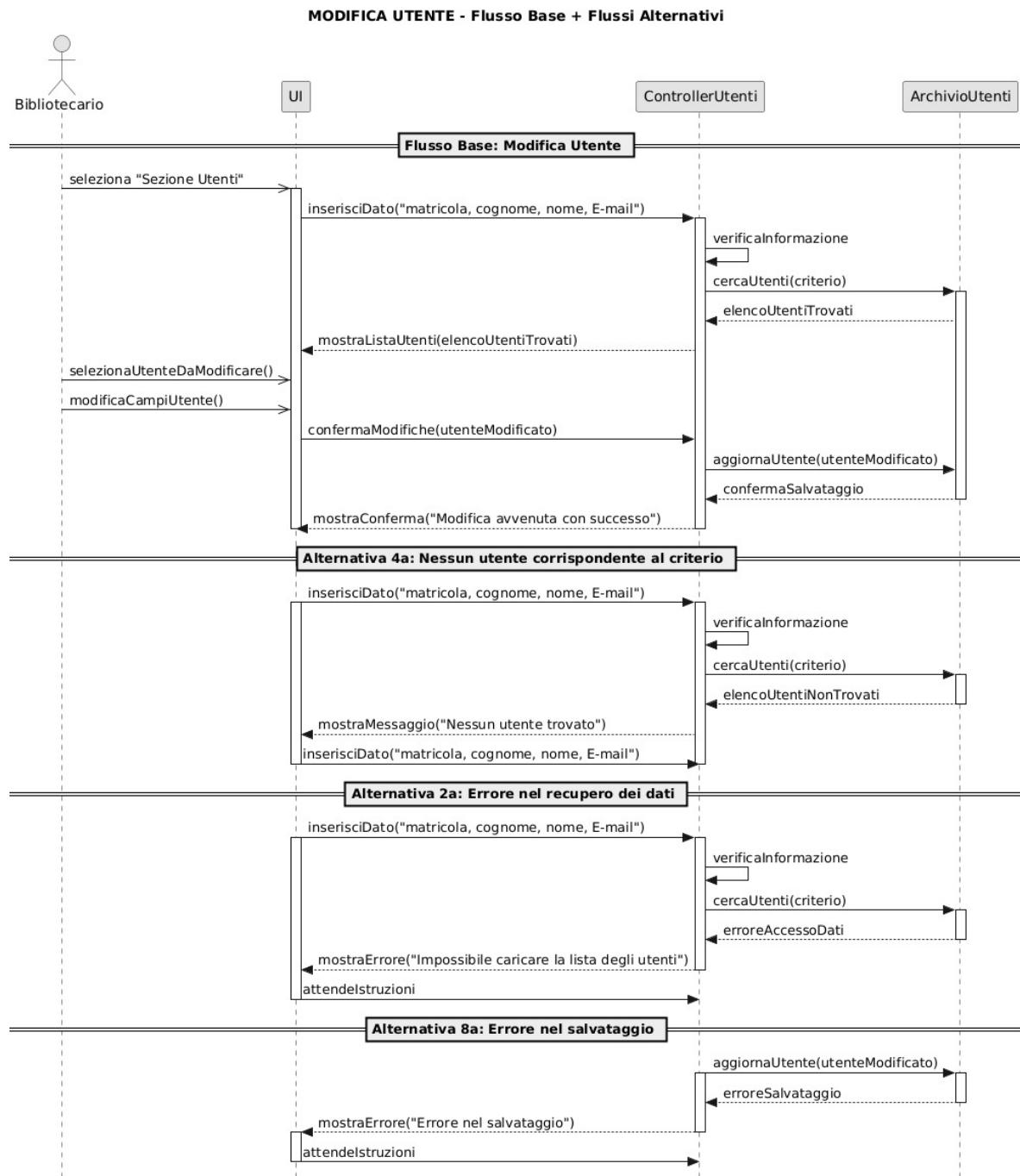


Figure 2.13: Diagramma di Sequenza — UC-13 Modifica utente

2.14 UC-14 — Elimina utente

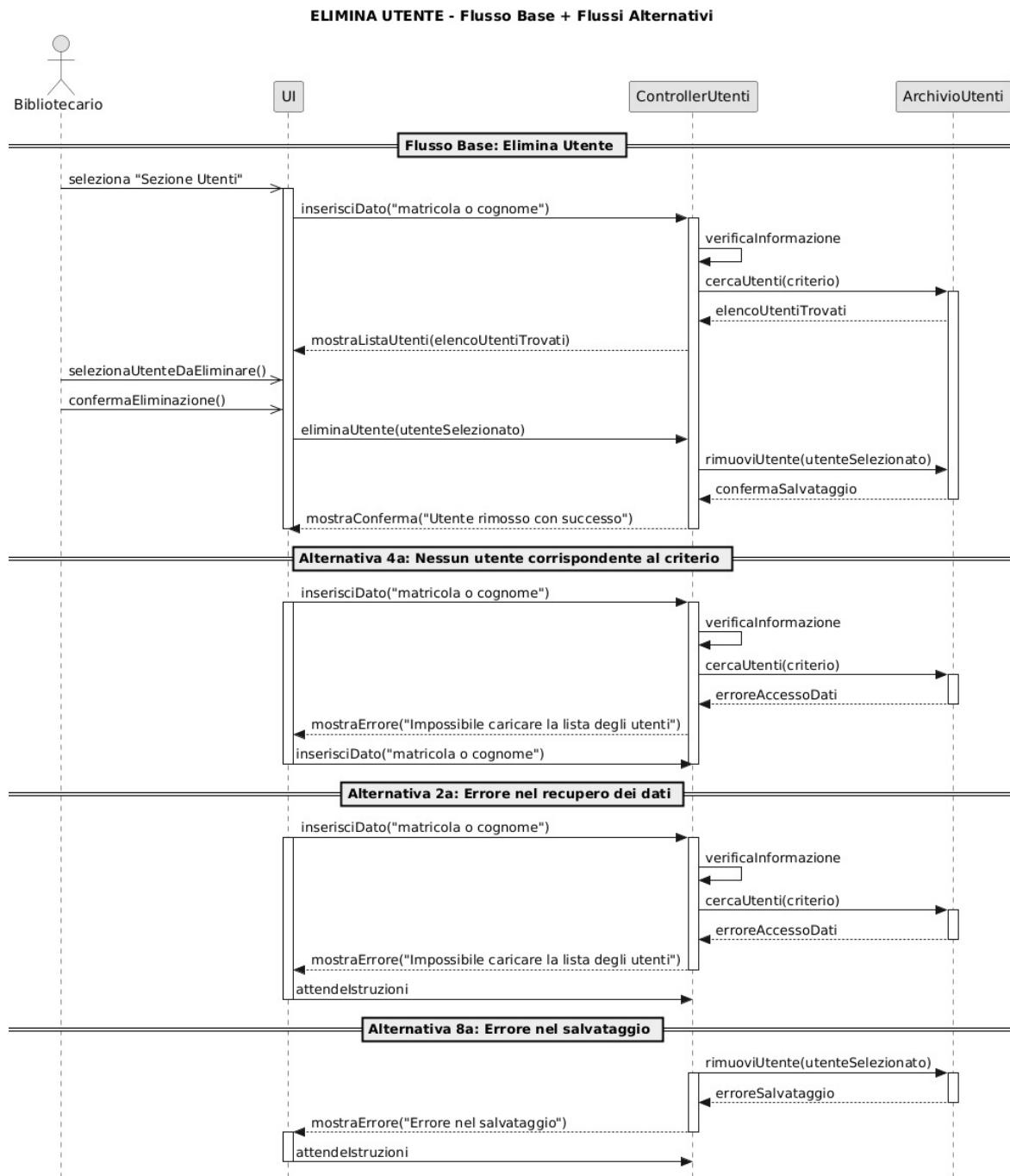


Figure 2.14: Diagramma di Sequenza — UC-14 Elimina utente

2.15 UC-15 — Inserisci utente

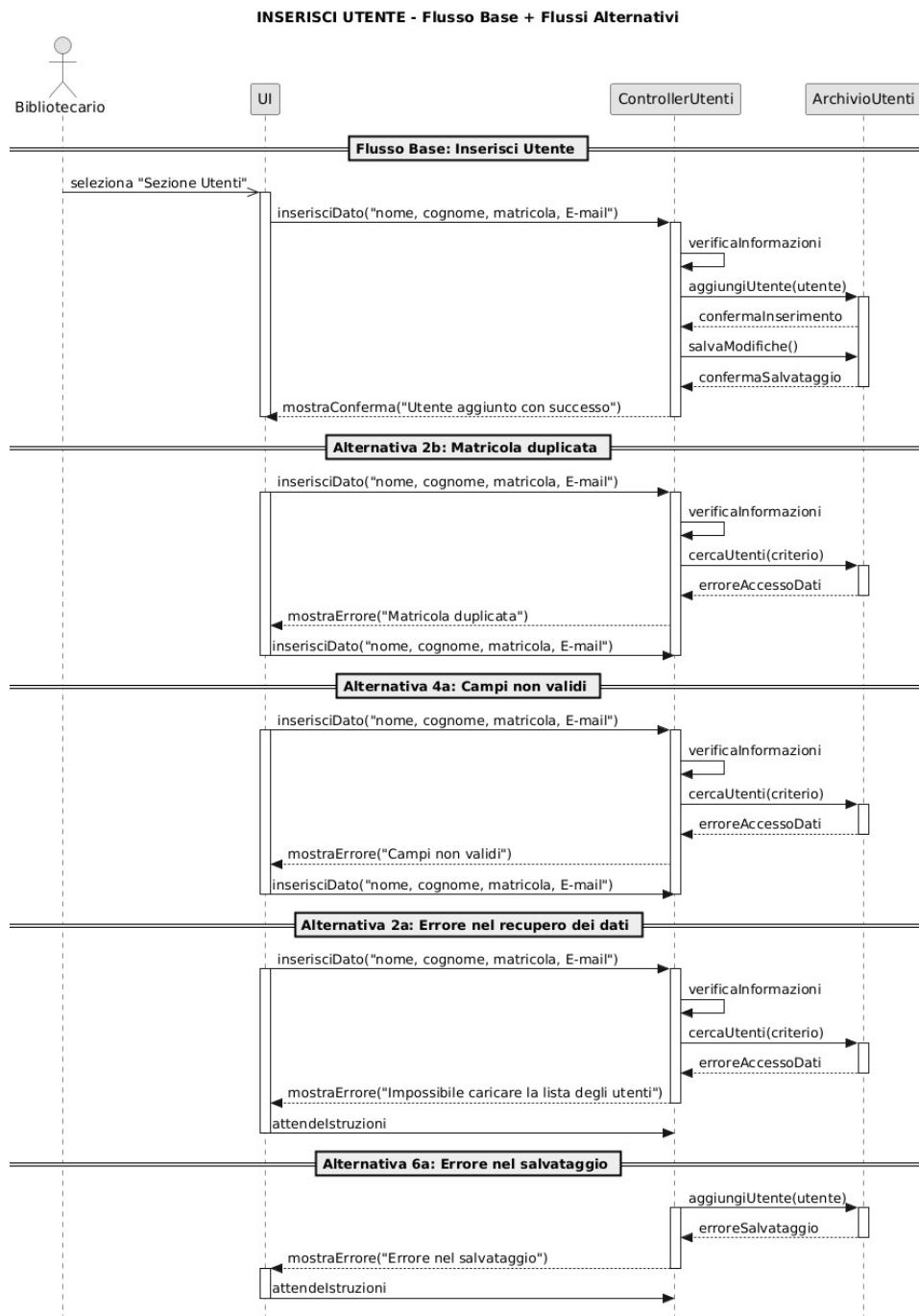


Figure 2.15: Diagramma di Sequenza — UC-15 Inserisci utente