

Progettazione Architetturale

1) Architettura del sistema

L'architettura del sistema è organizzata seguendo il pattern MVC (Model-View-Controller) unito ad un Layered pattern. La suddivisione in package non è soltanto una scelta estetica ma è pensata per garantire separazione dei ruoli, manutenibilità, scalabilità, facilità di estensione futura, alta coesione e basso accoppiamento.

Per quanto riguarda i package abbiamo:

- Package root

Il package root contiene la classe App, che rappresenta il punto d'ingresso dell'intera applicazione.

Oltre a occuparsi del caricamento delle interfacce grafiche tramite FXML, App è responsabile anche dell'inizializzazione dell'intero sistema, creando e collegando tra loro tutti i componenti fondamentali, tra cui i package service, data e controller.

- Package view

Il package view contiene tutti i file FXML che definiscono la parte grafica del sistema. Le view sono prive di logica: descrivono soltanto l'interfaccia utente (come layout, bottoni, campi di testo).

Questa scelta permette di mantenere separata grafica e logica, apportare modifiche all'interfaccia senza toccare il codice,

- Package controller

Il package controller raccoglie tutte le classi che si occupano dell'interazione tra l'interfaccia grafica e la logica applicativa sottostante. E' in grado di ricevere input dall'utente attraverso la View, validare i dati, invocare i metodi offerti dal servizio BibliotecaService, aggiornare la View sulla base dei risultati ottenuti.

- Package service

Il package service, contenente la classe BibliotecaService, che rappresenta la logica applicativa del sistema, fornisce la possibilità di inserire, modificare ed eliminare i dati relativi a Libri, Studenti e Prestiti

- Package data

Il package data è composto dalla classe BibliotecaData, che agisce come contenitore dello stato dell'applicazione e come struttura dati centrale.

Il Service e i Repository interagiscono con questo componente per salvare e caricare lo stato delle entità. Questa soluzione permette di evitare la

dispersione dello stato tra più parti del programma e garantisce un unico punto di gestione dei dati.

- **Package repository**

Il package repository gestisce tutto ciò che riguarda l'accesso ai dati. Composto dalle classi LibroRepository, StudenteRepository, PrestitoRepository, e dall'interfaccia Repository, contengono le strutture dati necessarie all'immagazzinamento, corrispettivamente dei libri, studenti e prestiti.

- **Package model**

Il package model è composto dalle 3 classi, Libro, Studente e Prestito, che rappresentano le entità principali del dominio.

L'architettura di questo sistema è stata organizzata in modo tale da ottenere una struttura chiara, facilmente estendibile e facilmente gestibile. E' stata suddivisa in package strutturati in modo tale da ottenere la massima separazione delle responsabilità, proprio per questo ogni componente dell'applicazione svolge un ruolo preciso e ben definito, evitando dipendenze inutili, e favorendo un'elevata coesione e un basso accoppiamento tra i componenti.

E' stata adottata un'organizzazione che favorisse la manutenibilità del codice e che rendesse semplice introdurre nuove funzionalità senza dover intervenire in più punti della struttura.

Un altro aspetto positivo di questa implementazione, è la testabilità. I vari packages possono essere testati separatamente

Pattern Architetturali utilizzati

1-MVC(Model View Controller)

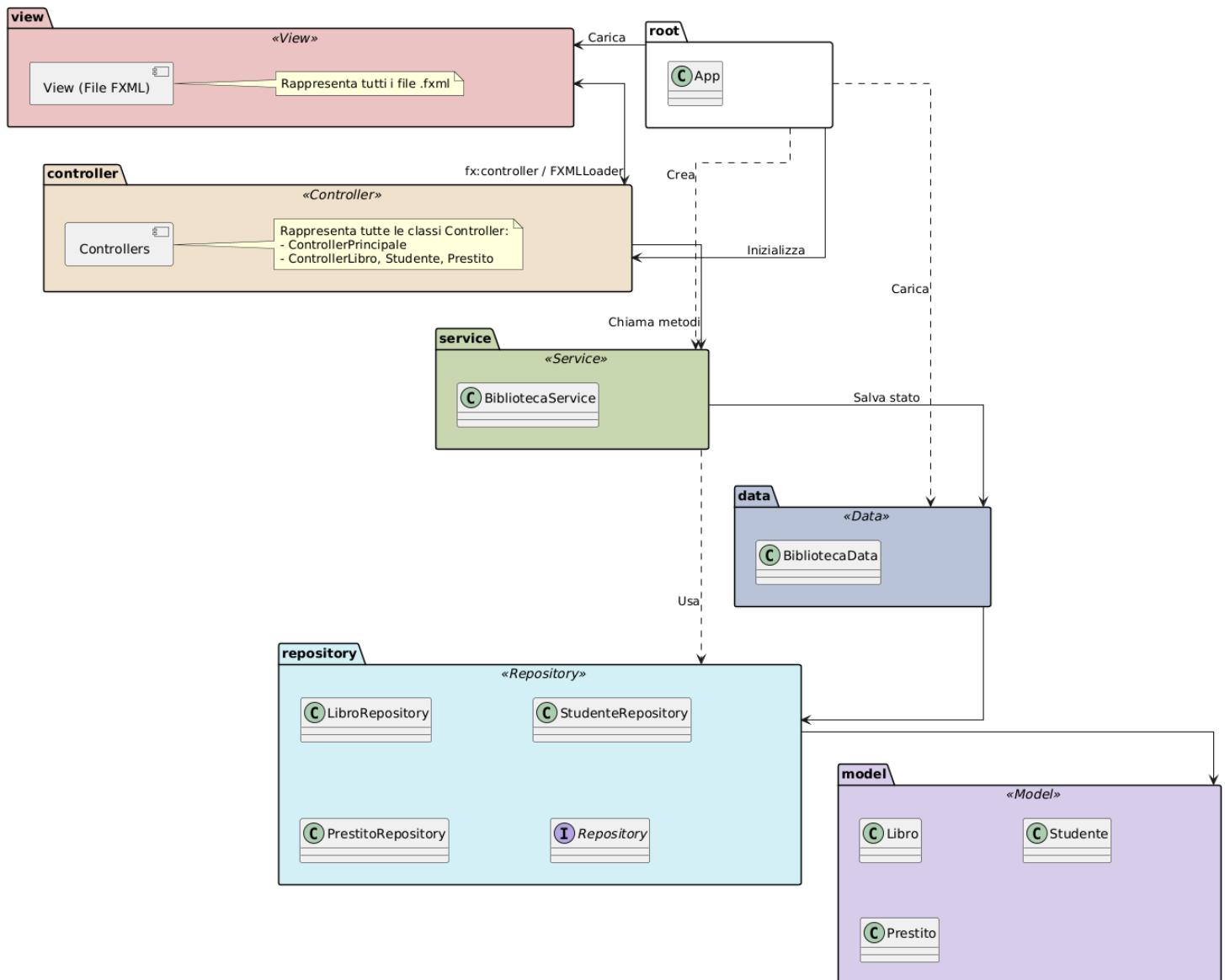
- Model: Contiene le entità del dominio, quali Libro, Studente e Prestito
- View: Gestisce la parte grafica, tramite FXML.
- Controller: Contiene la logica che mette in relazione Model e View

2-Layered Pattern

Sebbene l'interazione utente segua il pattern MVC (Model-View-Controller), l'architettura complessiva del sistema 'BibliotecaUniSa' è organizzata secondo una Layered Architecture stratificata, che garantisce modularità e separazione delle responsabilità:

- Presentation Layer (view): Costituito dal package view, contiene i file FXML che definiscono la struttura grafica delle interfacce.
- Application/Control Layer (controller/root): Rappresentato dal package controller, gestisce gli eventi dell'interfaccia utente e funge da intermediario, coordinando la vista ma delegando le operazioni complesse.
- Business Logic Layer (service): Incapsulato nel package service (classe BibliotecaService), rappresenta il cuore funzionale dell'applicazione dove risiede la logica di dominio.
- Persistence Layer (data & repository): Suddiviso nei package data e repository, si occupa dell'inizializzazione dei dati (BibliotecaData) e dell'astrazione dell'accesso alle entità (LibroRepository, StudenteRepository, PrestitoRepository) definite nel Model.

Diagramma Architettonico Sintetico - BibliotecaUniSa



2) Modello statico

Il modello statico mostra l'organizzazione del sistema o di una sua parte mostrando gli elementi principali e le loro relazioni senza ancora considerarne il comportamento nel tempo. In questa sezione si utilizzano i diagrammi di classe per definire la struttura statica delle classi di oggetti e le loro associazioni in un sistema software.

La progettazione di dettaglio usufruisce del modello statico per indicare con precisioni classi, metodi, attributi e le relazioni tra gli oggetti.

Nel realizzare il diagramma delle classi abbiamo adoperato dei principi generali di buona progettazione quali:

- DRY – Don't Repeat Yourself: evitando duplicazioni di attributi o metodi tra classi attraverso meccanismi di astrazione evitando così una manutenzione complessa e contraddizioni logiche.
- YAGNI: You Aren't Going to Need It: Realizzato solo metodi realmente necessari.
- Separation of Concerns: aspetti distinti del sistema sono gestiti da moduli diversi per garantire semplicità nello sviluppo e nella manutenibilità.
- Single Responsibility Principle(SRP): ogni componente del codice svolge un unico e specifico compito così che eventuali modifiche vengano effettuate solo nel modulo responsabile di quella funzionalità e non nell'intera classe.
- Principio di inversione della dipendenza: i moduli di alto livello non dipendono dai moduli di basso livello, ma dipendono entrambi da astrazioni così che sia il livello alto che quello basso riducono le loro dipendenze.

Questi principi favoriscono alta coesione in quanto le classi hanno responsabilità ben definite e basso accoppiamento per la dipendenza da astrazioni. Partendo dal fondo del nostro diagramma dei package, descriviamo i diagrammi delle classi realizzate.

Diagramma delle classi del package Model

Includere la classe Libro, Studente e Prestito. I “...” indicano la presenza dei getter e/o dei setter per tutti gli attributi nelle classi in cui appaiono, che sono stati omessi per semplificare la leggibilità e comprensibilità del diagramma. Nella classe Prestito sono presenti solo i setter mostrati mentre anche in questo caso sono state omesse le getter per tutti gli altri attributi. La classe prestito utilizza la relazione di composizione con le classi Libro e Studente perché non può esistere autonomamente. Non si può creare un prestito senza un libro valido invece il libro può esistere senza prestito. La cardinalità 1...1 indica che ogni prestito riguarda un

solo libro. Inoltre il prestito non può essere anonimo quindi è dipendente da studente che può avere massimo tre libri in prestito contemporaneamente come indicato dalla cardinalità 1..*.

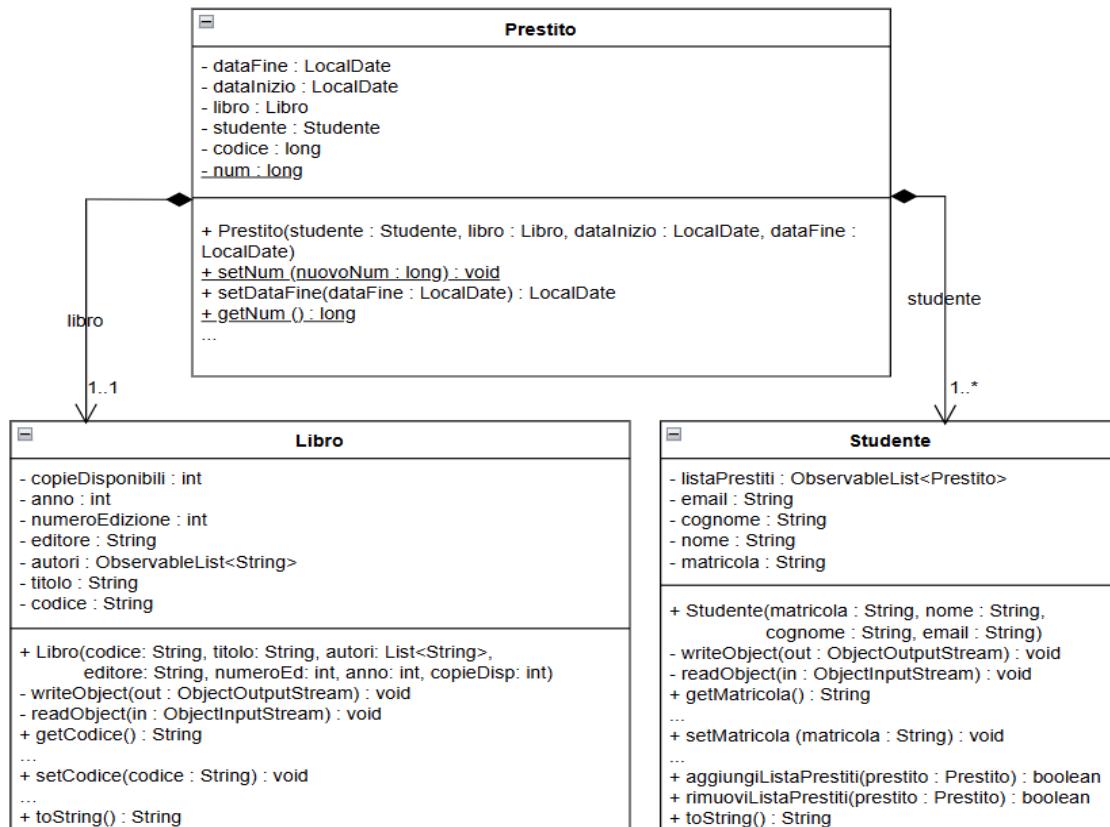


Diagramma delle classi del package Repository

include l'interfaccia Repository con dei metodi astratti che vengono implementati da tutte le classi che la ereditano, in linea con il principio di buona progettazione DRY. In tal caso, il principio è applicato alla firma dei metodi (nome e lista dei tipi di parametri) che vengono definiti una sola volta nell'interfaccia, e poi implementati nelle rispettive classi che li ereditano. La relazione presente è l'implementazione per collegare le classi presenti con l'interfaccia. Le classi LibroRepository, PrestitoRepository e StudenteRepository, oltre ai metodi ereditati, hanno due attributi: ObservableList e Map. L'ObservableList ha lo scopo di integrarsi al meglio con le TableView permettendo di aggiornare automaticamente l'interfaccia ogni volta che ci sono cambiamenti nella lista senza dover fare refresh manuali, mentre la Map viene utilizzata come indice parallelo alla Observable List per permettere una ricerca molto più veloce ($O(1)$) tramite codice identificativo dell'oggetto, evitando scansioni lente ($O(n)$) della lista. Con queste due collection si riesce ad avere una UI reattiva (grazie alla lista osservabile) e un accesso ai singoli elementi molto più efficiente e veloce.

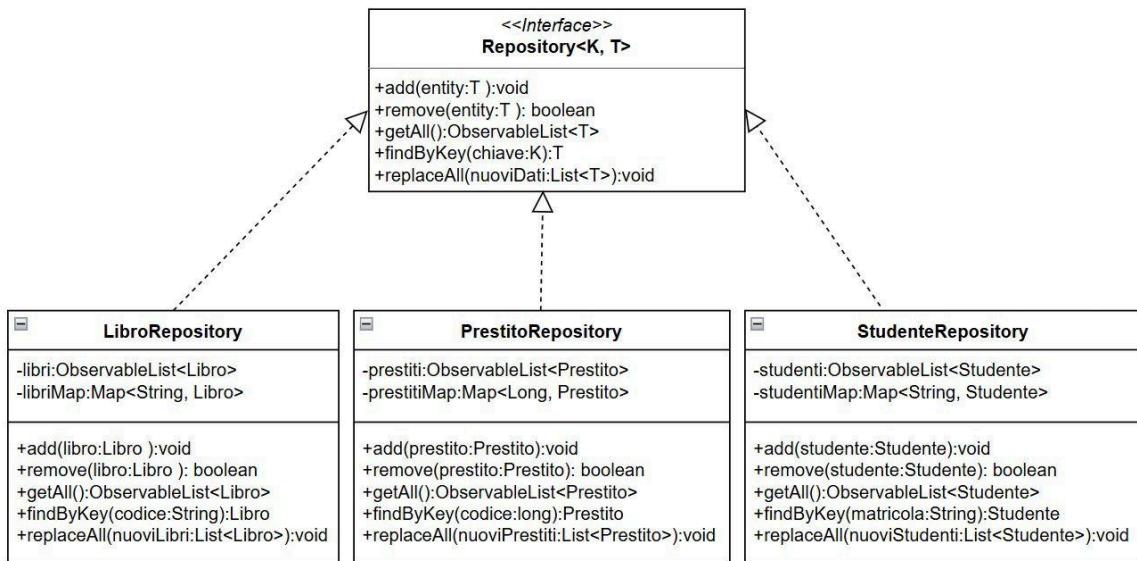


Diagramma delle classi del package Service

Il package Service contiene la classe

BibliotecaService che si occupa delle operazioni tra i vari repository e fornisce i metodi pubblici per gestire libri, studenti e prestiti .In accordo con Separation of Concerns e SRP si occupa solo delle operazioni logiche che sono quindi separate dalla gestione dei dati. Offre funzionalità come l'aggiunta, la modifica, la rimozione, la ricerca e l'accesso alla lista osservabile dei libri e degli studenti. Inoltre possiede metodi relativi ai prestiti, quali creazione e ricerca di un nuovo prestito, restituzione di un libro e visione dei prestiti attivi e archiviati. Nella classe BibliotecaService c'è un attributo data di tipo BibliotecaData utilizzato per recuperare i vari repository (libri, studenti, prestiti).

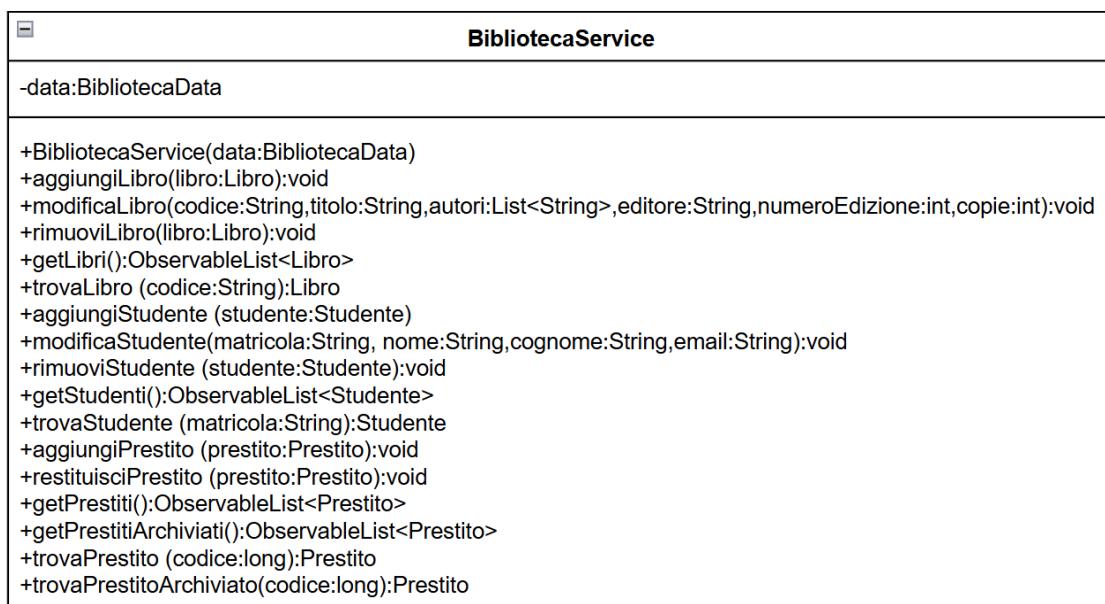


Diagramma delle classi del package Data

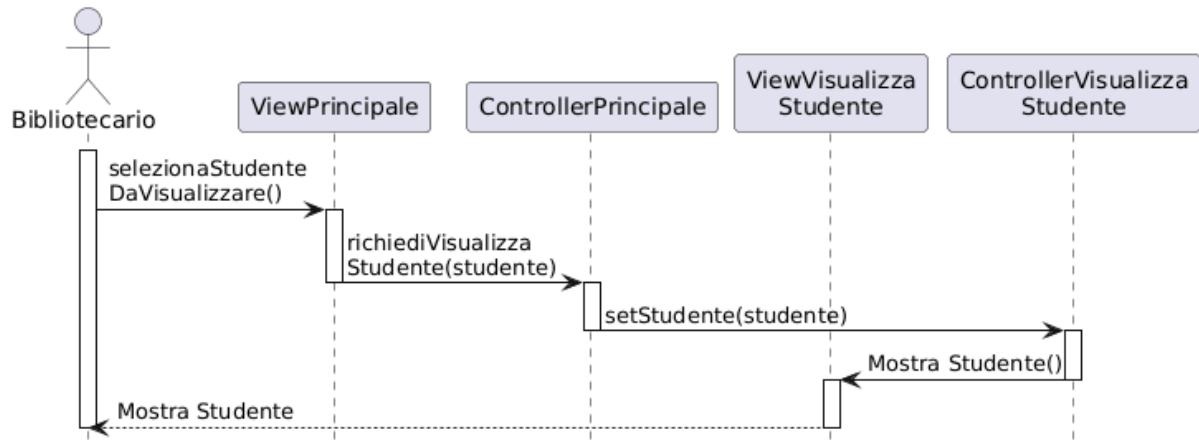
Il diagramma delle classi del package Data contiene la classe BibliotecaData. In linea con Separation of Concerns si occupa della gestione dei dati. Inoltre la classe ha un ruolo ben definito in accordo con SRP, si occupa della gestione dei tre repository fondamentali: LibroRepository, StudenteRepository e PrestitoRepository accessibili tramite i metodi getter. BibliotecaData permette di salvare e caricare lo stato dell'applicazione sia con percorso predefinito che con percorso specifico garantendo di non perdere i dati tra una sessione e l'altra. Inoltre è presente l'attributo DEFAULT_PATH che fornisce un salvataggio predefinito.

BibliotecaData
<p>-DEFAULT_PATH:String -libroRepo:LibroRepository -studenteRepo:StudenteRepository -prestitoRepo:PrestitoRepository</p>
<p>+BibliotecaData() +getLibroRepo():LibroRepository +getStudenteRepo():StudenteRepository +getPrestitoRepo:PrestitoRepository +salvaStato():void +salvaStato (filePath:String):void +caricaStato ():void +caricaStato (filePath:String):void</p>

3. Modello dinamico

In questa fase di progettazione si scende nell'analisi del comportamento del sistema e di come le componenti descritte nel modello statico interagiscono tra loro. I diagrammi più usati in questa fase sono i diagrammi di sequenza che vengono modellati sulla base dei casi d'uso descritti in precedenza nel documento di specifica dei requisiti.

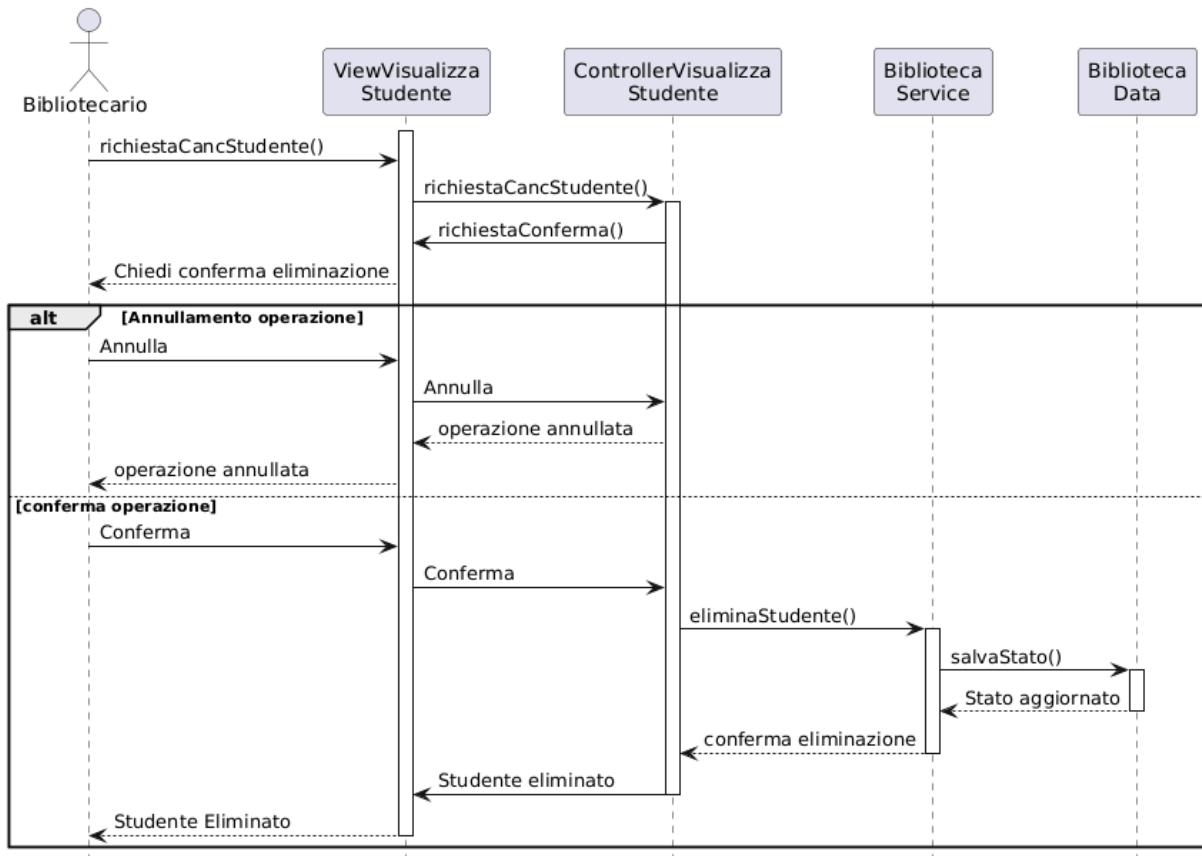
DS1: Visualizzazione



Riprende il caso d'uso della visualizzazione, nel diagramma si parla nello specifico dell'operazione di Visualizzazione studente, ma il susseguirsi delle azioni e delle classi coinvolte è analogo per tutte le visualizzazioni (libri, prestiti e prestiti archiviati). Il bibliotecario tramite la View e il Controller principale, richiede la visualizzazione dello studente selezionato dalla tabella. Il ControllerPrincipale prende le informazioni dello studente richiesto (già disponibili senza dover consultare le repository) e le passa al controller VisualizzaStudente, che a sua volta le invia alla ViewVisualizzaStudente. Quest'ultima permette al bibliotecario di visualizzare nel dettaglio le informazioni sullo studente selezionato.

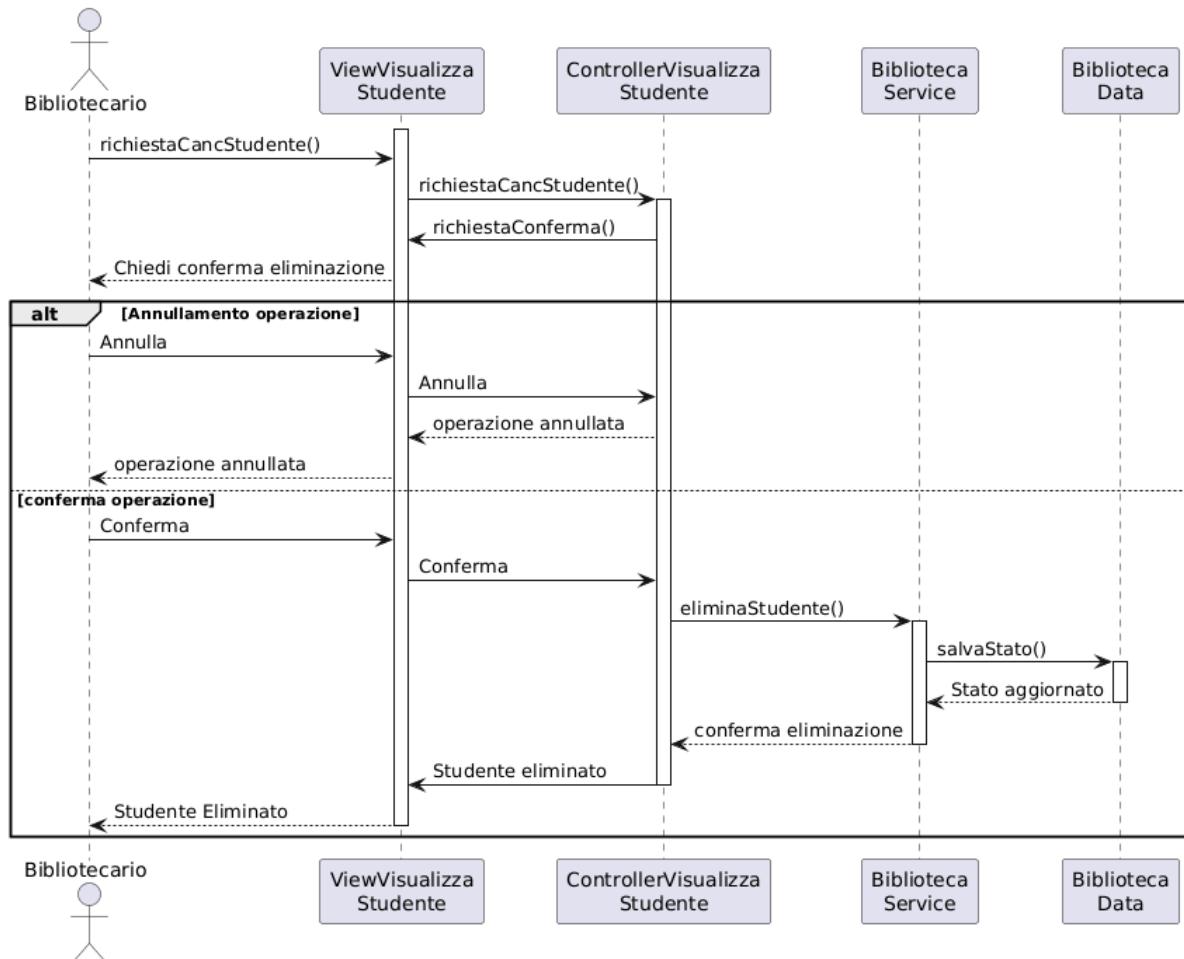
Come già accennato prima, in questa fase non c'è bisogno di accedere alla StudenteRepository per ottenere le informazioni dello studente richiesto, perché la lista aggiornata degli studenti è già stata caricata in precedenza, permettendo così al bibliotecario di poter visualizzare i dati.

DS2: Modifica



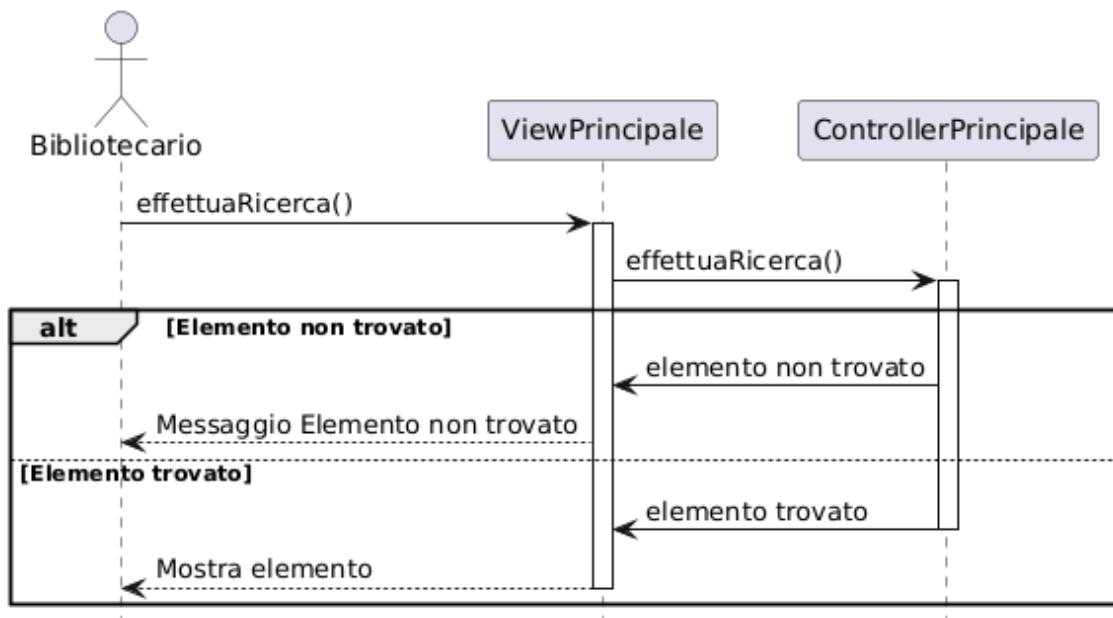
Riprende il caso d'uso dell'operazione modifica Studente, ma il diagramma può essere usato per descrivere anche l'operazione di modifica Libro. Come pre-condizione si prevede che il bibliotecario si trovi nella pagina di visualizzazione dello studente che si desidera modificare. Il bibliotecario chiede di poter effettuare la modifica e inizia a modificare i campi. Se i campi non rispettano un certo tipo, viene notificato al bibliotecario e gli viene impedito di proseguire finchè non saranno inseriti correttamente. Nel momento in cui i campi sono corretti, il loop si interrompe e non viene inviato il messaggio di errore. L'operazione prosegue aggiornando Biblioteca Data tramite Biblioteca Service con le nuove informazioni fornite e il processo si termina notificando al Bibliotecario che l'operazione è andata a buon fine.

DS3: Cancella



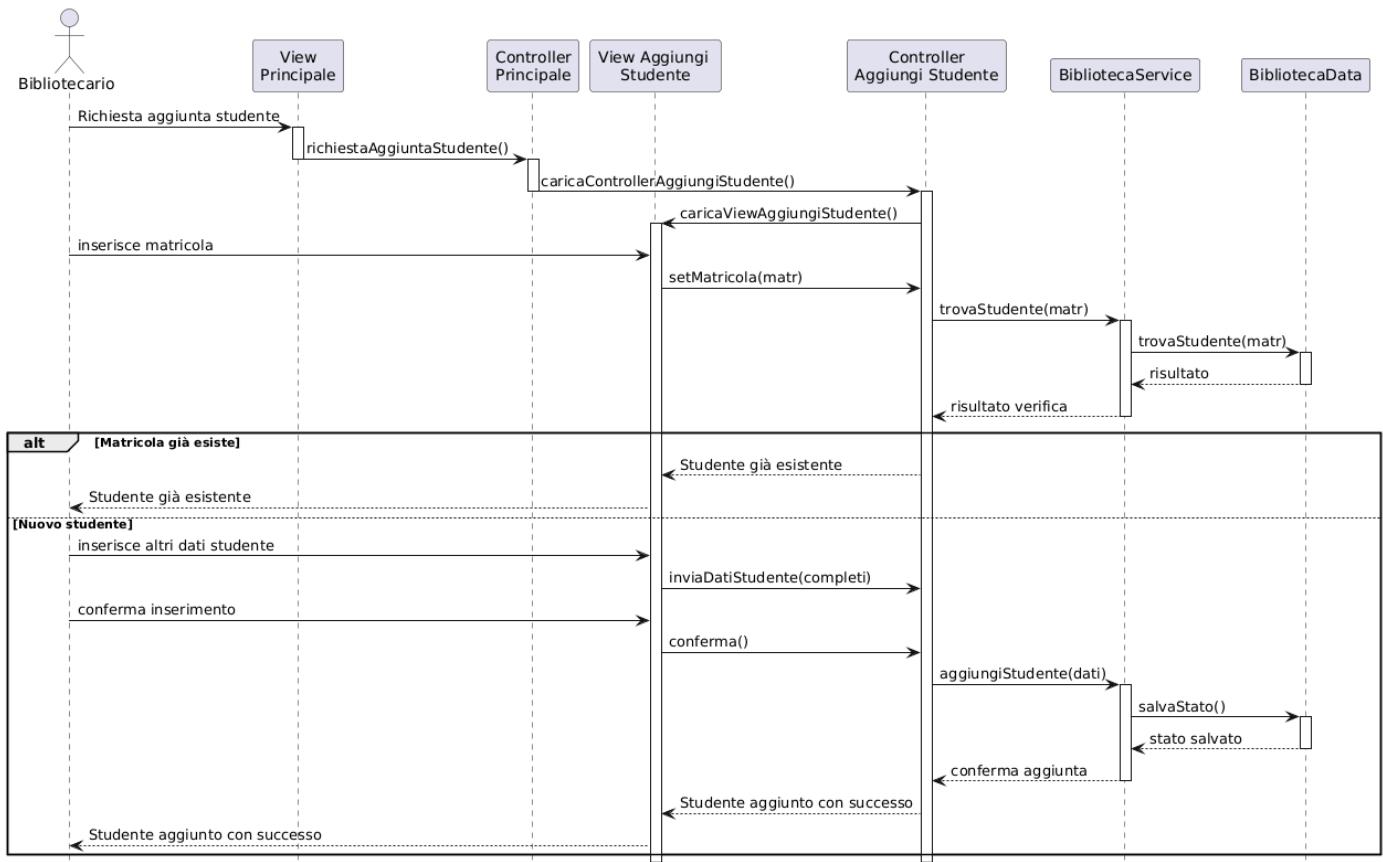
Riprende il caso d'uso della cancellazione studente, ma il processo è il medesimo per la cancellazione libro. Come pre-condizione si prevede che il bibliotecario si trovi nella pagina di visualizzazione dello studente che si desidera cancellare. Il bibliotecario richiede di poter effettuare la cancellazione dello studente e il Controller risponde tramite la view richiedendo la conferma dell'operazione. Il bibliotecario può scegliere di annullare l'operazione e mantenere invariata BibliotecaData, oppure confermare l'operazione e procedere all'operazione. A quel punto il Controller invoca il metodo di eliminaStudente di Biblioteca Service, che elimina lo studente e salva il nuovo stato con BibliotecaData. Al bibliotecario ritorna il messaggio di avvenuta eliminazione.

DS4: Ricerca



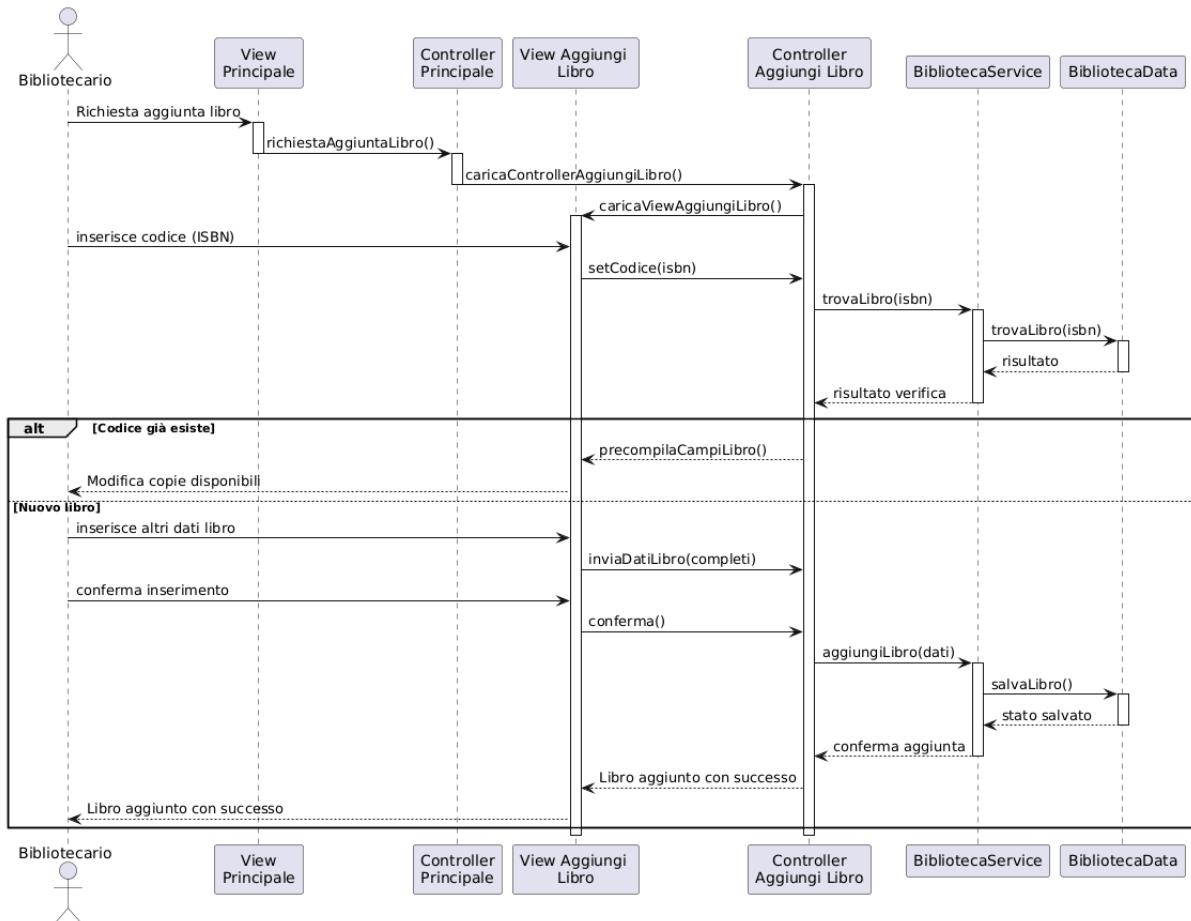
Riprende tutti i casi d'uso riguardanti la ricerca delle informazioni di studenti, libri, prestiti attivi o archiviati. Il bibliotecario inserisce nella barra di ricerca le informazioni che vuole ricercare (Attenzione! le informazioni inserite si presuppone siano inerenti al tipo di oggetto che si sta cercando in quella pagina: Se mi trovo nella tab degli studenti, allora inserirò informazioni come nome, cognome ecc). Le informazioni inserite vengono passate al Controller Principale che effettua la ricerca e restituisce un messaggio di errore in caso di ricerca non andata a buon fine, oppure i risultati della ricerca più inerenti alle info richieste. Come nel caso della Visualizzazione (DS1) non vi è necessità di accedere alle Repository apposite per effettuare la ricerca.

DS5: Registra Studente



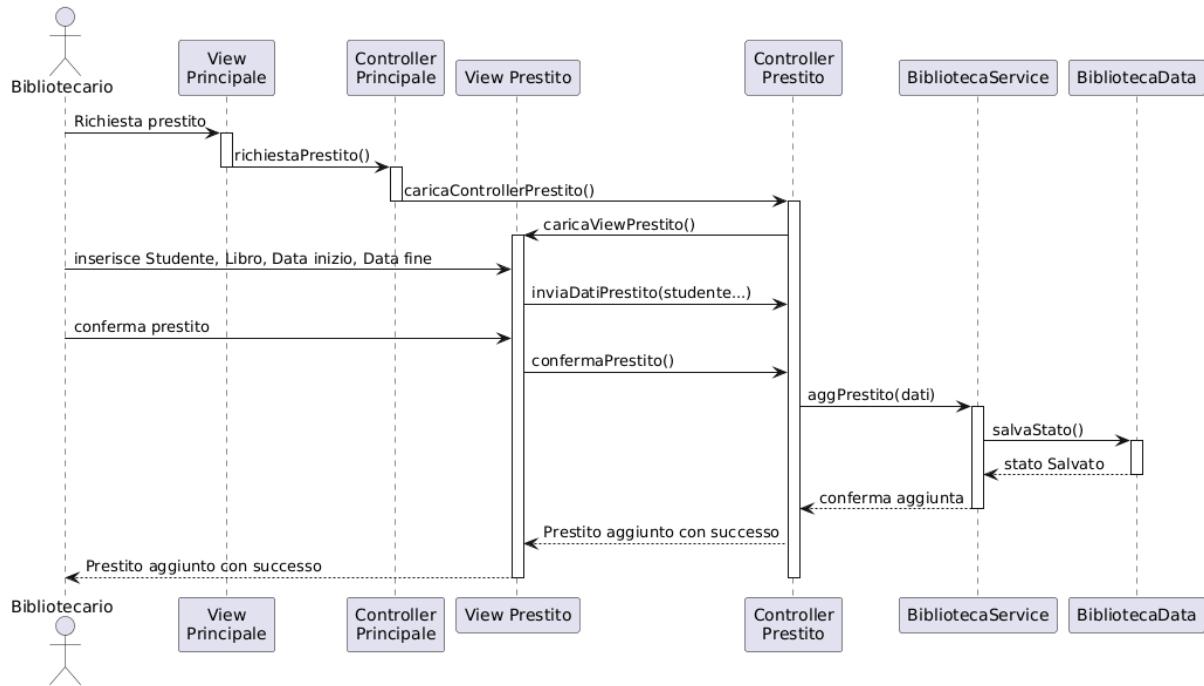
Riprende il caso d'uso della registrazione studente. Dalla view principale, il Bibliotecario richiede al controller di aggiungere uno studente. Il ControllerPrincipale invia la richiesta al Controller Aggiungi Studente che attiva la View Aggiungi Studente. Il Bibliotecario inserisce poi i campi, primo fra tutti la matricola che viene confrontata con quelle presenti nel sistema. Se il sistema trova nell'archivio la matricola, allora notifica che lo studente è già stato registrato, in caso contrario il bibliotecario può proseguire con l'inserimento dei campi e poi confermare l'operazione. Lo studente viene poi registrato e inserito nel Sistema tramite Biblioteca Data. Il sistema notifica poi al bibliotecario tramite la view Aggiungi studente che l'operazione è andata a buon fine.

DS6: Registra Libro



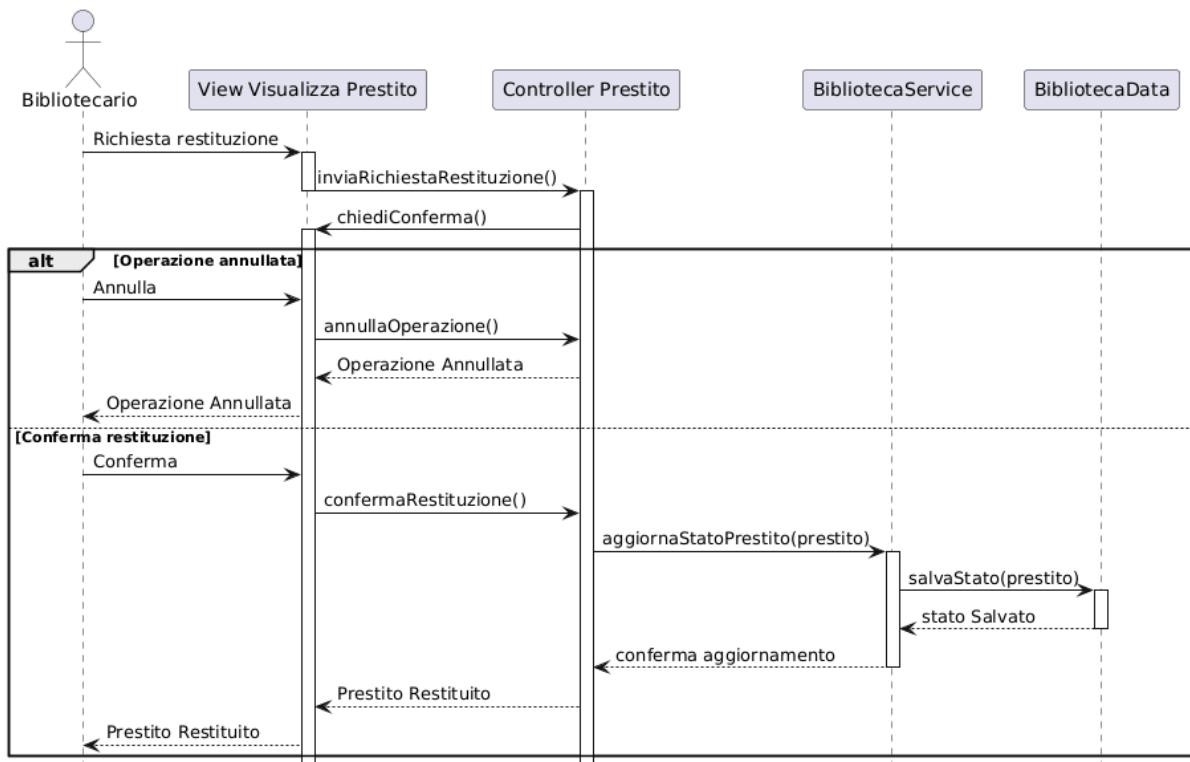
Riprende il caso d'uso della registrazione libro. Dalla view principale, il Bibliotecario richiede al controller di aggiungere un libro. Il ControllerPrincipale invia la richiesta al ControllerAggiungiLibro che attiva la ViewAggiungiLibro. Il Bibliotecario inserisce il codice ISBN che viene confrontato con quelli già presenti nel sistema. Se il sistema trova nell'archivio il codice, allora notifica che il libro è già presente nel sistema, compila in automatico gli altri campi con le informazioni corrette e permette al bibliotecario la modifica del numero di copie. Nel caso in cui il codice non venga trovato, il bibliotecario può proseguire con l'inserimento dei campi e poi confermare l'operazione di registrazione. Il libro viene poi registrato e inserito nel Sistema tramite Biblioteca Data. Il sistema notifica infine al bibliotecario tramite la View Aggiungi Libro che l'operazione è andata a buon fine.

DS7: Prestito



Riprende il caso d'uso della registrazione prestito. Dalla view principale, il Bibliotecario richiede al controller di registrare un prestito. Il ControllerPrincipale invia la richiesta al ControllerPrestito che attiva la ViewPrestito. Il Bibliotecario inserisce lo studente che richiede il prestito, il libro da prestare e la data di inizio e fine prestito previsto. Lo studente e il libro vengono inseriti utilizzando la ricerca: vengono cercati nel sistema lo studente e libro corrispondente, selezionati e confermati per effettuare il prestito. Una volta inseriti i dati, il Bibliotecario manda tramite ViewPrestito la conferma dell'operazione. Il prestito viene poi registrato e inserito nel Sistema tramite Biblioteca Service e Biblioteca Data. Il sistema notifica infine al bibliotecario tramite la ViewPrestito che l'operazione è andata a buon fine.

DS8: Restituzione



Riprende il caso d'uso della restituzione prestito. Come pre-condizione si prevede che il bibliotecario si trovi nella pagina di visualizzazione del prestito che si desidera restituire. Il bibliotecario richiede di poter effettuare la restituzione del libro e il ControllerPrestito risponde tramite la View richiedendo la conferma dell'operazione. Il bibliotecario può scegliere di annullare l'operazione e mantenere invariata BibliotecaData, oppure confermare l'operazione e procedere. A quel punto il Controller invia alla BibliotecaService la richiesta di aggiornare lo stato eliminando il prestito che verrà poi inserito nei prestiti archiviati. BibliotecaService inoltra a BibliotecaData la richiesta di salvataggio Stato e al bibliotecario ritorna il messaggio di avvenuta restituzione.

Nota: Alla fine di tutti i diagrammi di sequenza, a seguito della notifica di Successo o completamento dell'operazione, vengono impostati la view e il controller principali sulla rispettiva tab da cui si era partiti.

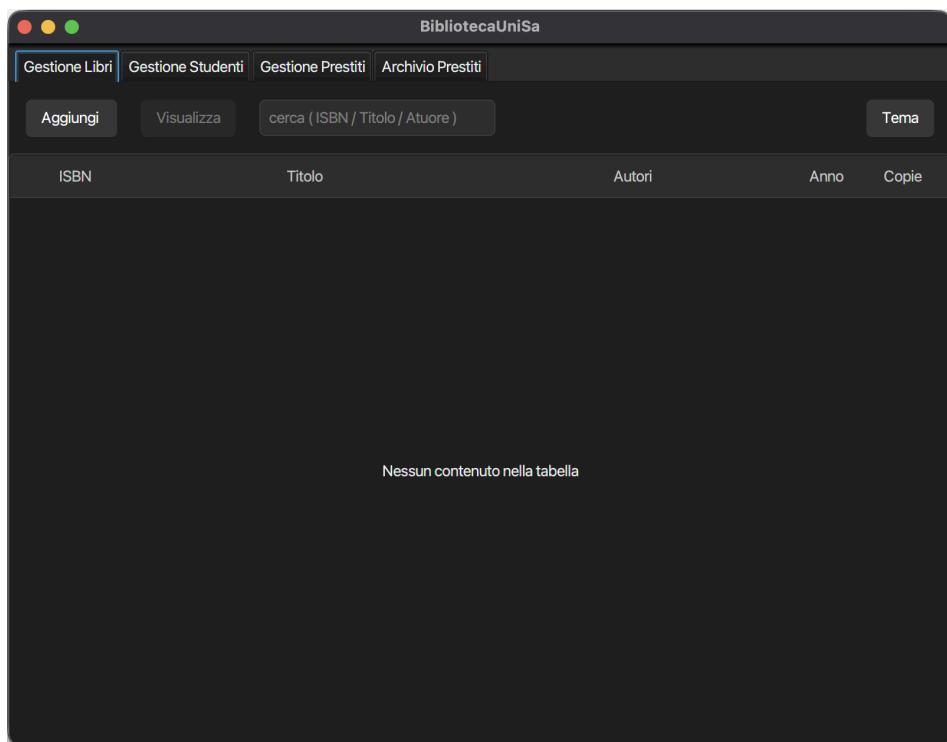
4. Design dell'interfaccia utente

Includere eventuali mock-up delle schermate principali.

Descrivere i mock-up illustrando le principali interazioni previste con l'utente.

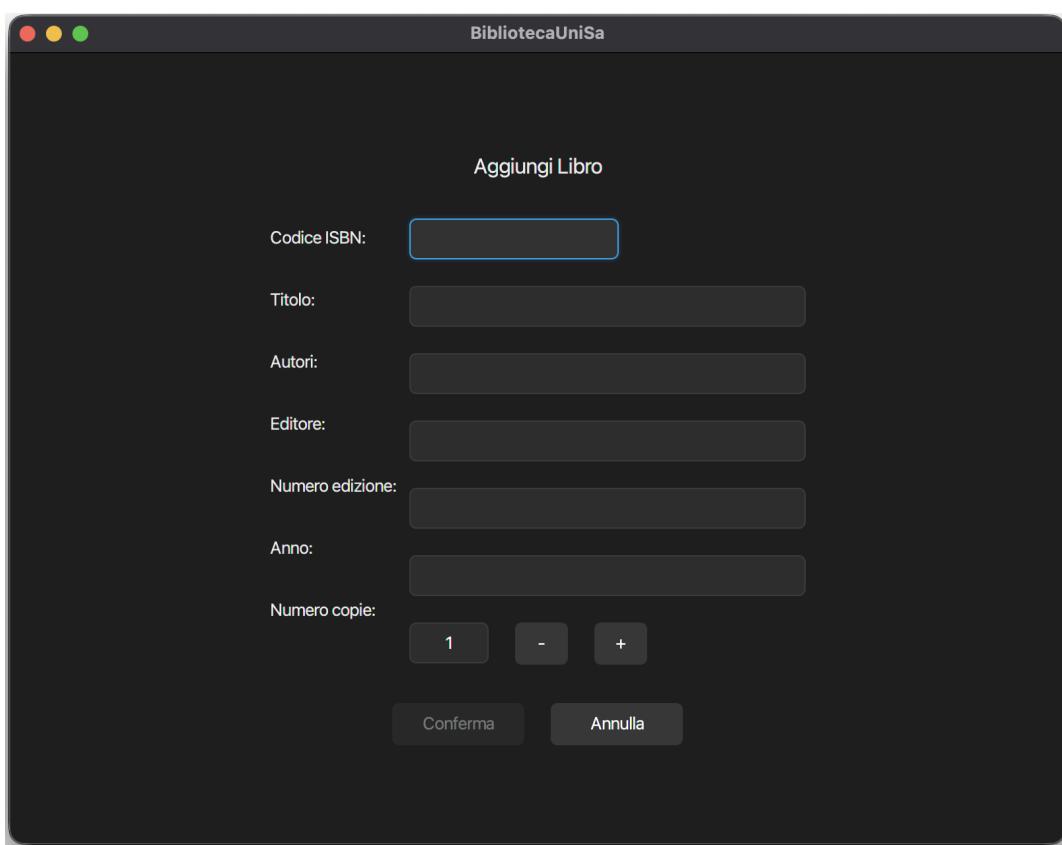
1) Gestione Libri

Questa è la schermata principale, che appare all'avvio del programma. Tramite questa finestra è possibile aggiungere nuovi libri cliccando sull'apposito tasto ‘Aggiungi’, filtrare le ricerche in base ad ISBN, Titolo, o Autore digitando nell'apposito box ‘cerca (ISBN / Titolo / Autore)’, visualizzare l'elenco dei libri nell'apposita tabella, che se vuota restituisce a schermo ‘Nessun contenuto nella tabella’. E' poi possibile ordinare questa lista, cliccando sui corrispettivi tasti, per ISBN, Titolo, Autori, Anno o Copie, in base alle proprie preferenze. Nel caso in cui la tabella avesse almeno un elemento, cliccandoci sopra e premendo successivamente sul tasto ‘Visualizza’ (il quale resta non cliccabile finché non viene selezionato un elemento della tabella), vedremo maggiori informazioni relative all'elemento selezionato.



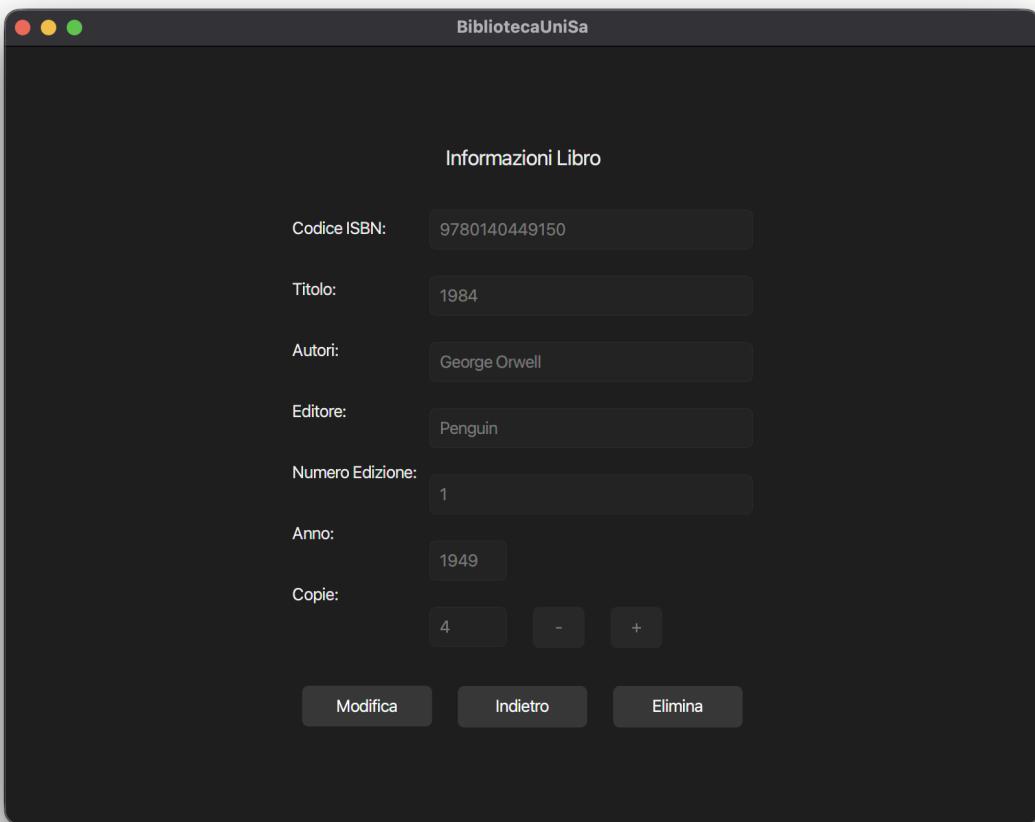
1.1) Aggiungi Libro

Dopo aver cliccato sul tasto ‘Aggiungi’, nella sezione Gestione Libri, comparirà la seguente schermata. Bisognerà compilare correttamente i campi ‘Codice ISBN’ (13 cifre), ‘Titolo’, ‘Autori’, ‘Editore’, ‘Numero edizione’, ‘Anno’, ‘Numero copie’. Per questo ultimo campo è possibile impostare il numero tramite interfaccia grafica, cliccando su ‘+’ per aumentare di 1 il numero di copie, ‘-’ per diminuirlo (il sistema impedirà di cliccarlo se il numero di copie è pari ad 1). Soltanto dopo aver compilato tutti i campi, sarà possibile cliccare sul tasto ‘Conferma’ per poter inserire il libro nella lista. Al contrario per il tasto ‘Annulla’, cliccabile sin dall’inizio, che interromperà il processo di inserimento del libro nella lista, chiudendo la schermata e ritornando a quella di Gestione Libri.



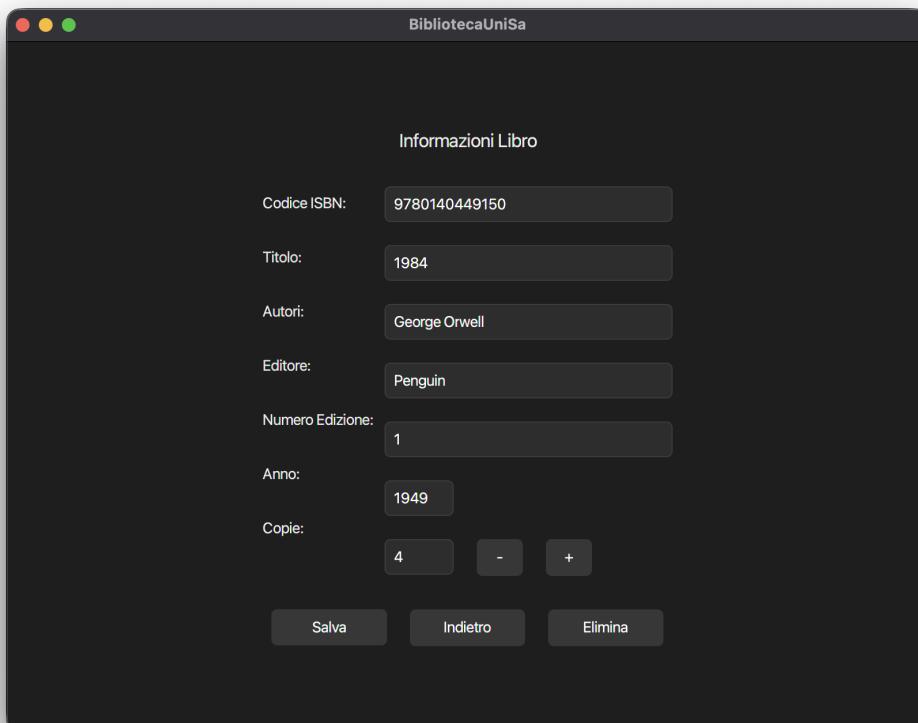
1.2) Visualizza

Dopo aver selezionato il libro dalla lista, cliccando su ‘Visualizza’, appare la seguente schermata. Il sistema permette di visualizzare in sola lettura tutte le informazioni relative al libro. Cliccando sul tasto ‘Modifica’, è possibile modificare tutti i campi tranne il ‘Codice ISBN’. Il tasto ‘Indietro’ permette di tornare alla schermata precedente, il tasto ‘Elimina’ permette di eliminare il libro dalla lista dei libri.



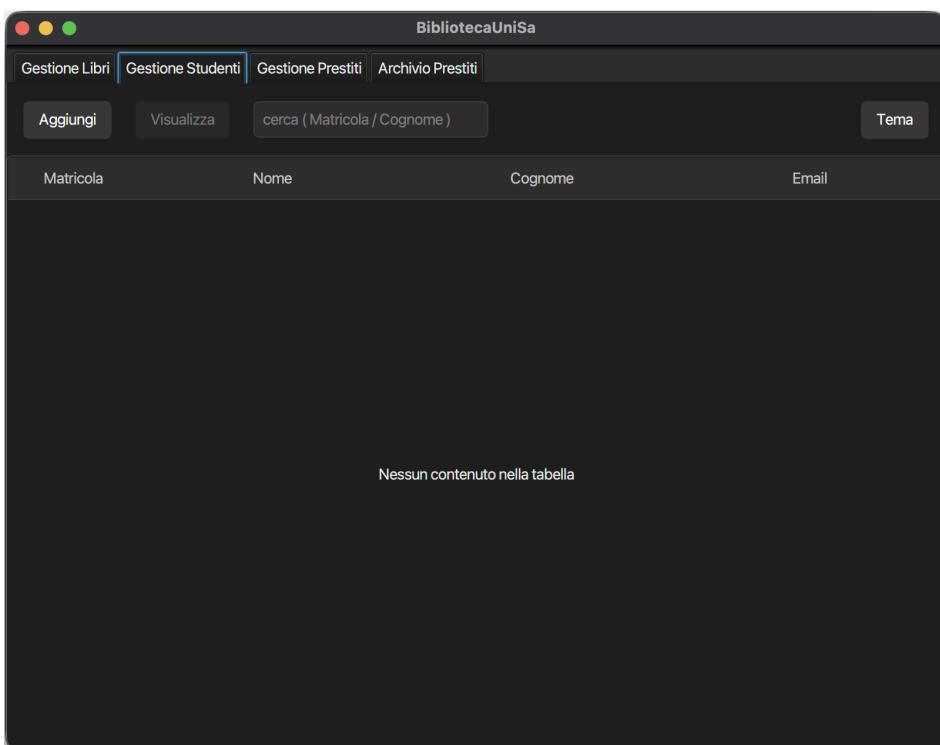
1.2.1) Modifica

Dopo aver cliccato il tasto ‘Modifica’, il sistema ci consente di modificare i campi ‘Titolo’, ‘Autori’, ‘Editore’, ‘Numero Edizione’, ‘Anno’, ‘Copie’. Per salvare le modifiche, cliccare il tasto ‘Salva’. ‘Indietro’ permette di tornare alla schermata precedente, ‘Elimina’ di eliminare il libro dalla lista.



2) Gestione Studenti

In questa schermata avviene la gestione degli studenti: come per i Libri, è possibile aggiungerli cliccando sul tasto Aggiungi, filtrare la ricerca tramite l'apposito campo ‘cerca (Matricola / Cognome)’. Nel caso in cui la lista contenesse almeno uno studente, questo comparirebbe nella tabella. Nel caso contrario, compare a schermo la scritta a ‘Nessun contenuto nella tabella’. E’ possibile poi ordinare la lista degli Studenti in base alle proprie preferenze, cliccando corrispettivamente su ‘Matricola’, ‘Nome’, ‘Cognome’ o ‘Email’. Nel caso in cui la tabella avesse almeno un elemento, cliccandoci sopra e premendo successivamente sul tasto Visualizza (cliccabile solo dopo aver selezionato un elemento della tabella), vedremo maggiori informazioni relative all’elemento selezionato.



2.1) Aggiungi Studente

Cliccando sul tasto Aggiungi nella schermata Gestione Studenti, apparirà la seguente finestra. Per aggiungere uno studente, bisogna compilare i campi ‘Matricola’ (10 cifre), ‘Nome’, ‘Cognome’, ‘Email’. Soltanto dopo aver compilato correttamente tutti i campi sarà possibile cliccare sul tasto ‘Conferma’. E’ invece sempre possibile annullare l’inserimento cliccando sul tasto ‘Annulla’, che ti riporterà alla schermata di Gestione Studenti.

BibliotecaUniSa

Aggiungi Studente

Matricola:

Nome:

Cognome:

Email:

2.2 Visualizza

Dopo aver selezionato lo studente dalla lista, cliccando sul tasto ‘Visualizza’, apparirà la seguente schermata. L’interfaccia è suddivisa in due parti. La prima, quella superiore, in cui vengono specificate le informazioni riguardanti lo studente, in sola lettura, (Matricola, Nome, Cognome, Email), la seconda, in cui vengono specificati i prestiti relativi a tale studente (già in corso). Il sistema dà la possibilità di modificare le informazioni relative allo studente cliccando sul tasto ‘Modifica’, ad eccezione della matricola , la quale NON può essere modificata. Il tasto ‘Elimina’, ci permette di cancellare lo studente dalla lista degli studenti. Il tasto ‘Indietro’ invece permette di tornare alla schermata Gestione Studenti.

BibliotecaUniSa

Informazioni Studente

Matricola: 1000000001

Nome: Mario

Cognome: Rossi

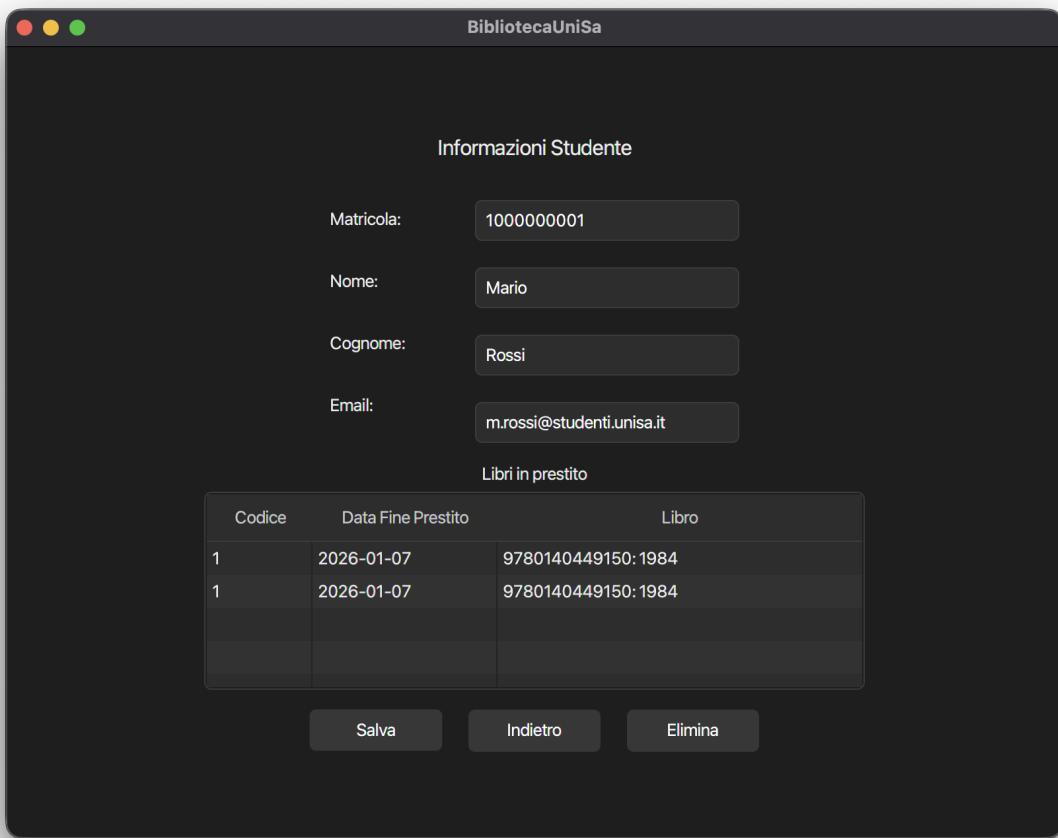
Email: m.rossi@studenti.unisa.it

Libri in prestito

Codice	Data Fine Prestito	Libro
1	2026-01-07	9780140449150: 1984
1	2026-01-07	9780140449150: 1984

2.2.1) Modifica

Dopo aver cliccato sul tasto ‘Modifica’, il sistema consente la modifica dei campi ‘Nome’ , ‘Cognome’ , ‘Email’. Per salvare le modifiche, cliccare il tasto ‘Salva’. ‘Indietro’ permette di tornare alla schermata precedente, ‘Elimina’ di eliminare lo studente dalla lista.



Informazioni Studente

Matricola: 1000000001

Nome: Mario

Cognome: Rossi

Email: m.rossi@studenti.unisa.it

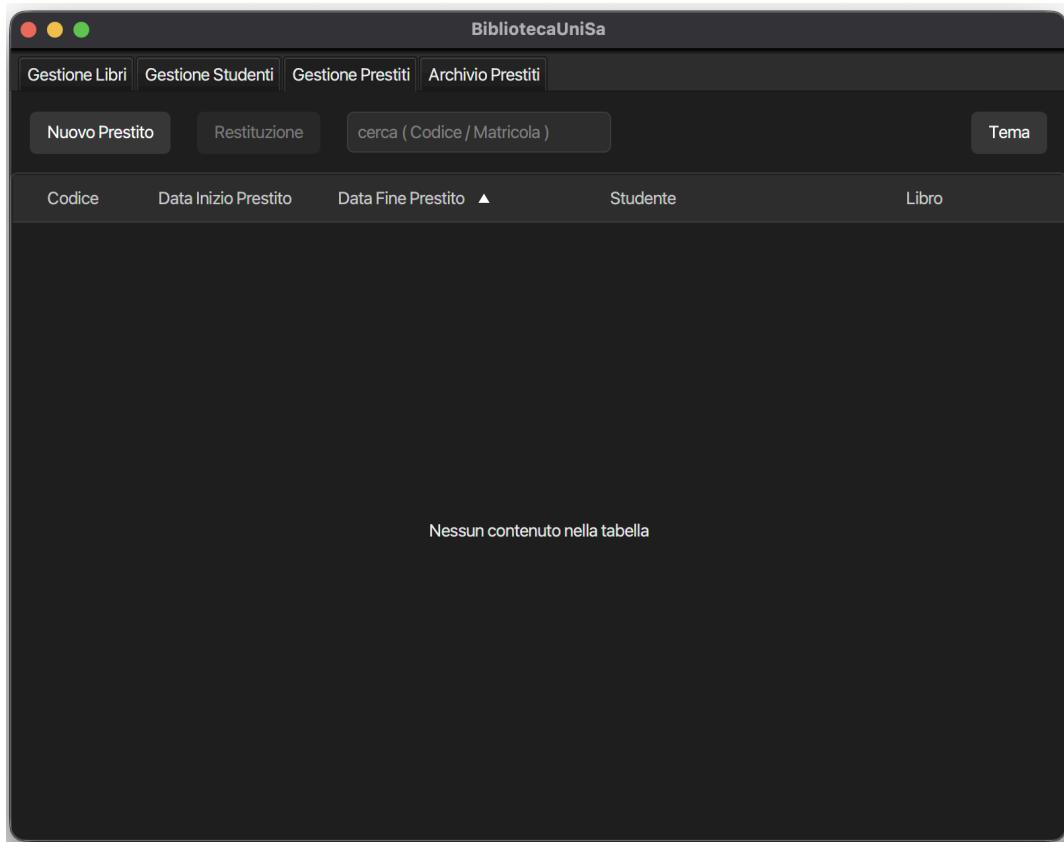
Libri in prestito

Codice	Data Fine Prestito	Libro
1	2026-01-07	9780140449150:1984
1	2026-01-07	9780140449150:1984

Salva Indietro Elimina

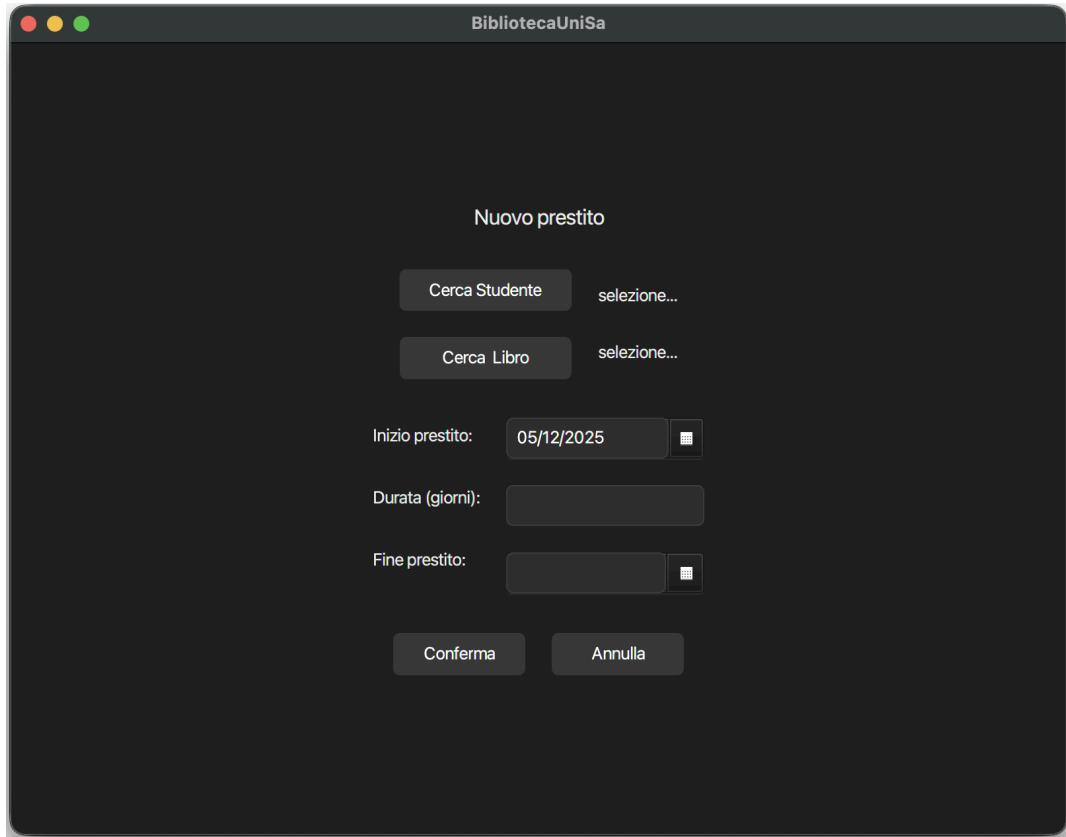
3) Gestione prestiti

In questa sezione, è possibile gestire tutto ciò che riguarda i prestiti. E’ possibile aggiungere un prestito cliccando sull’apposito tasto ‘Nuovo Prestito’, filtrare la lista dei prestiti digitando il filtro nella sezione ‘cerca (Codice / Matricola)’, ordinarli cliccando corrispettivamente su ‘Codice’, ‘Data Inizio Prestito’, ‘Data Fine Prestito’, ‘Studente’ , ‘Libro’. E’ possibile inoltre registrare la restituzione del prestito cliccando sull’apposito tasto ‘Restituzione’.



3.1) Nuovo Prestito

Cliccando sul tasto ‘Nuovo Prestito’ in Gestione Prestiti, apparirà la seguente schermata. Per registrare un prestito, cliccando sul tasto ‘Cerca Studente’, il sistema ci fornirà la lista degli utenti registrati alla piattaforma. Selezionare lo studente che richiede il prestito. Cliccando su ‘Cerca Libro’, il sistema fornirà la lista di libri registrati sul sistema. Selezionare il libro richiesto dallo studente. Nel campo ‘Inizio prestito’, è possibile selezionare la data dell’inizio del prestito, di default inizializzata alla data del giorno corrente. Nella sezione ‘Durata Giorni’, inserire il numero di giorni della durata del prestito. La sezione ‘Fine Prestito’, non è modificabile direttamente. La data di restituzione viene calcolata in automatico in base al giorno di inizio prestito, e la sua durata. Cliccando su ‘Conferma’ , il prestito viene registrato in piattaforma. Cliccando ‘Annulla’ , i dati inseriti non vengono salvati e si ritorna alla schermata di Gestione Prestiti.



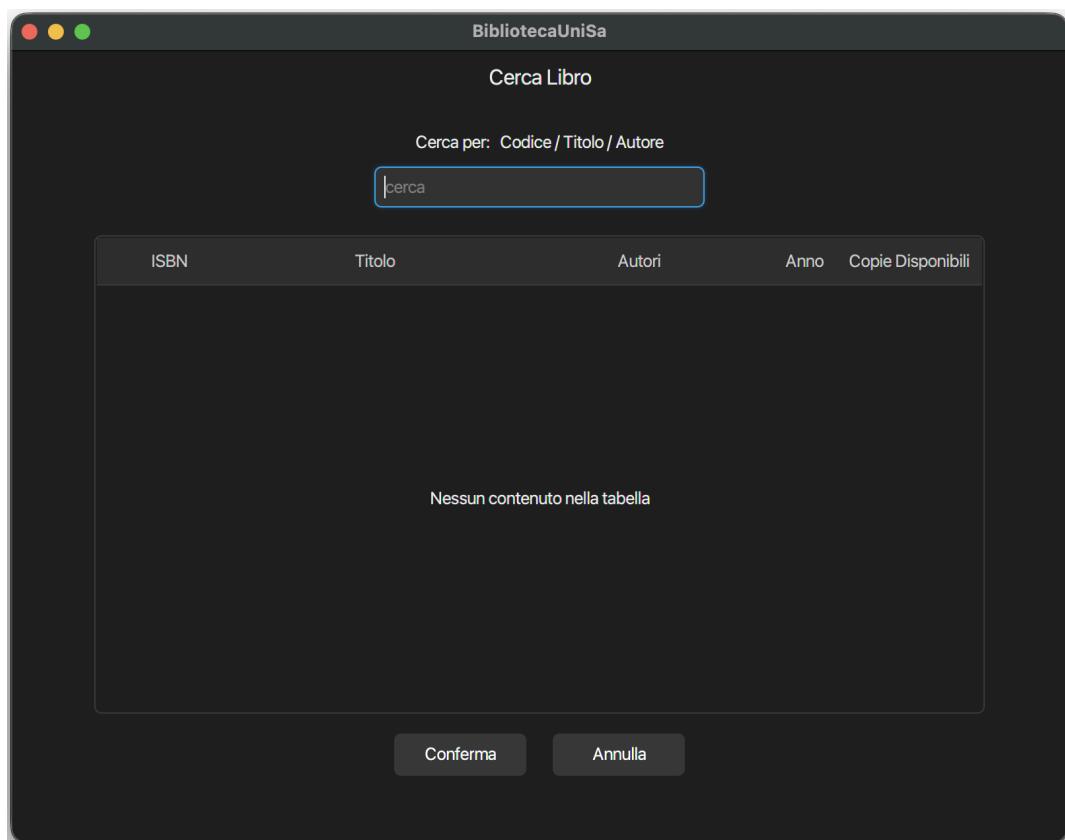
3.1.1)Cerca Studente

Dopo aver cliccato su ‘Cerca Studente’ in ‘Nuovo Prestito’ compare la seguente schermata. Questa mostra la lista di studenti registrati alla piattaforma. Bisogna selezionare lo studente interessato, e poi cliccare ‘Conferma’. Si ritornerà alla schermata precedente, in cui possiamo continuare a compilare i restanti campi. Digitando nel campo ‘cerca’, è possibile filtrare la lista per Matricola, Nome o Cognome per facilitare la ricerca di un determinato studente.

BibliotecaUniSa			
Cerca Studente			
Cerca per: Matricola / Nome / Cognome			
Matricola	Nome	Cognome	Email
Nessun contenuto nella tabella			
Conferma		Annulla	

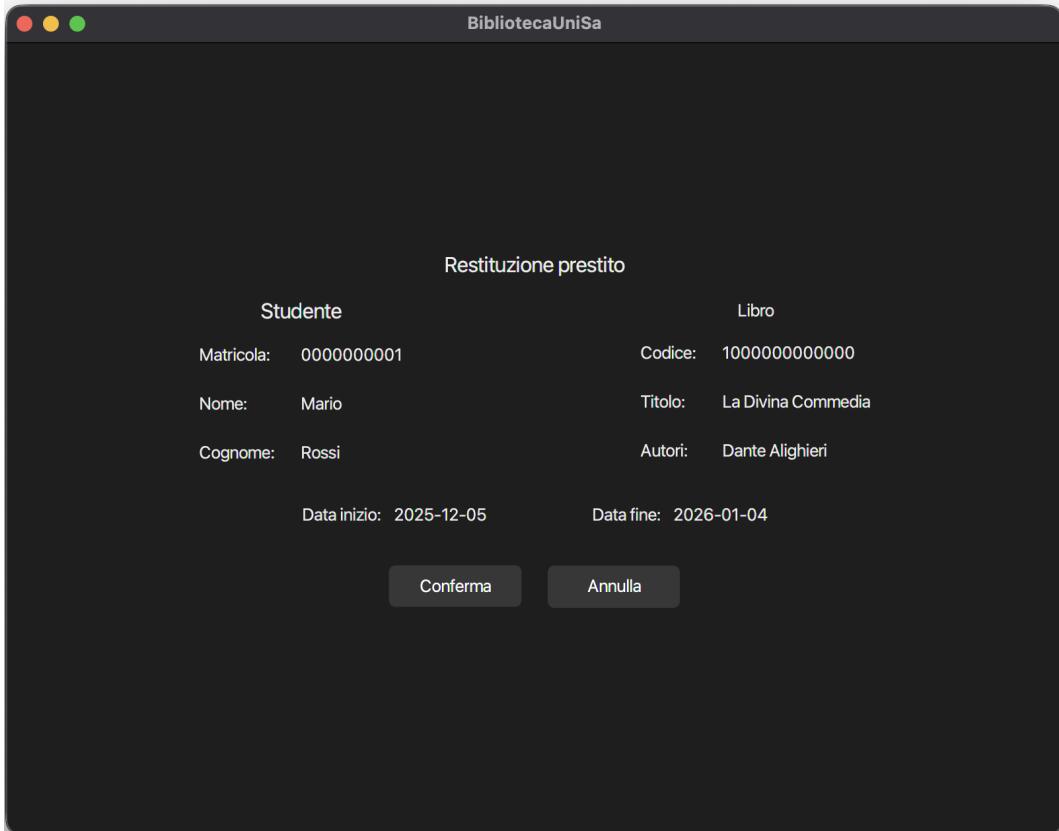
3.1.2) Cerca Libro

Dopo aver cliccato su ‘Cerca Libro’ in ‘Nuovo Prestito’, compare la lista di libri registrati sulla piattaforma. Bisogna selezionare il libro interessato, e cliccare successivamente su ‘Conferma’. Torneremo alla schermata precedente, con il campo ‘Cerca Libro’ compilato. Nel caso in cui premessimo invece ‘Annulla’, torneremmo comunque alla schermata precedente, ma con il campo non compilato. Digitando nel campo ‘cerca’, è possibile filtrare la lista per Codice, Titolo o Autore per facilitare la ricerca di un determinato libro.



3.4) Restituzione Prestito

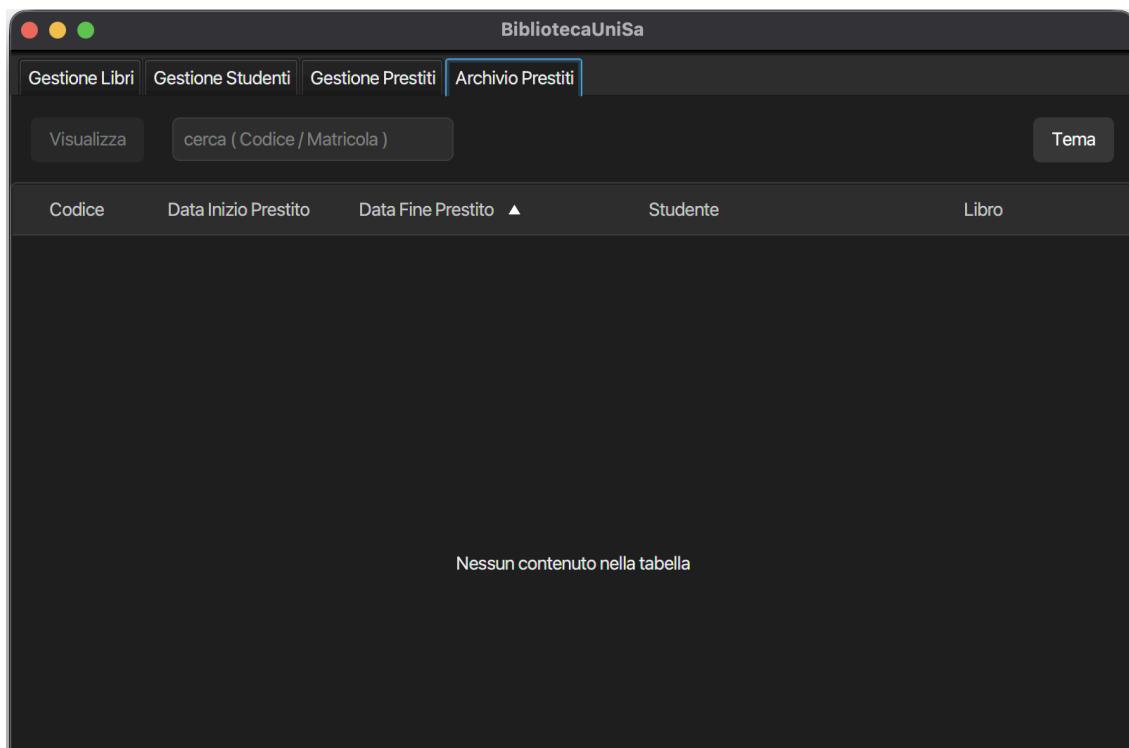
Dopo aver selezionato un prestito dalla lista, Cliccando sul tasto ‘Restituzione’, comparirà la seguente schermata, in cui compare un sommario di tutte le informazioni necessarie a stabilire che quello fosse effettivamente il prestito da restituire. E’ possibile cliccare ‘Conferma’ per confermare ed archiviare la restituzione del prestito, ‘Annulla’ per tornare indietro, senza salvare la restituzione del prestito.



4) Archivio Prestiti

In questa sezione dell'applicazione, viene riportato un elenco aggiornato automaticamente di tutti i prestiti restituiti, in modo tale da tenere traccia di tutto lo storico dei prestiti.

Il tasto ‘Visualizza’, cliccabile soltanto dopo aver selezionato un determinato prestito, restituisce maggiori informazioni su di esso. Il campo di testo ‘cerca (Codice / Matricola)’ consente di filtrare l’elenco, facendo comparire soltanto prestiti pertinenti a ciò che viene digitato in tale campo.



4.1) Visualizza

Dopo aver selezionato un prestito dall'elenco, premendo il tasto 'Visualizza' compare la seguente schermata. Questa ci permette di visualizzare tutte le informazioni relative a tale prestito, senza la possibilità di apportare alcuna modifica. Premendo il pulsante 'Indietro', il sistema torna alla schermata Archivio Prestiti.

