

Documento di Progettazione: Gestione Biblioteca Universitaria

Autori: Sabrina Soriano, Mirko Montella, Achille Romano, Ciro Senese

Gruppo: 02

Data: 8 novembre 2025

1. Architettura del sistema

Descrizione Generale:

L'applicazione è un software per la gestione di una biblioteca universitaria sviluppato in **Java** e con l'utilizzo di **JavaFX** per creare l'interfaccia grafica.

Per la sua implementazione abbiamo utilizzato il pattern architetturale MVC (Model-View Controller).

Lo utilizziamo perché sono disponibili più modi per visualizzare e interagire con i dati:

Es.1

- **View 1 (Studente/Guest):** Lo studente sfoglia la lista dei libri senza visualizzare le opzioni di gestione di tale lista.
- **View 2 (Bibliotecario):** Il bibliotecario visualizza la *stessa* lista di libri, ma arricchita con le opzioni per gestirla. Una riga con pulsanti per **"Modifica"** e **"Elimina"** e vede dettagli operativi (es. storico dei prestiti di quel libro).

Grazie a questo modello possiamo fare ciò non dovendo creare due database diversi (essendo il model è uno solo) ma semplicemente usando due View diverse in base al login.

Es. 2

- **View 1 (Lista):** Mostra i risultati come un elenco testuale formato da titolo, autore e ISBN.
- **View 2 (Griglia):** Mostra i risultati come una griglia di immagini (copertine dei libri).

Grazie a questo modello possiamo anche cambiare il modo in cui appaiono i risultati (griglia o lista) sempre aggiungendo semplicemente due View senza dover cambiare la ricerca che viene effettuata nel model.

Moduli Principali (Packages):

Model: (It.unisa.diem.ing.biblioteca.model)

- **Responsabilità:** Rappresenta il nucleo dell'applicazione perché contiene le classi principali che rappresentano e gestiscono i dati.
- **Classi principali:** App, Libro, Studente, Prestito, Bibliotecario, Database.
- **Dipendenze:** N/A

View: (it.unisa.diem.ing.biblioteca.app e resources.Biblioteca)

- **Responsabilità:** Contiene tutte le classi che rappresentano l'interfaccia grafica (UI).

Questo modulo si occupa solo di mostrare i dati e catturare gli input dell'utente •

Componenti principali: File primary.fxml, secondary.fxml, classe App.java. • **Dipendenze:**

Utilizza i Controller per delegare la gestione degli eventi utente.

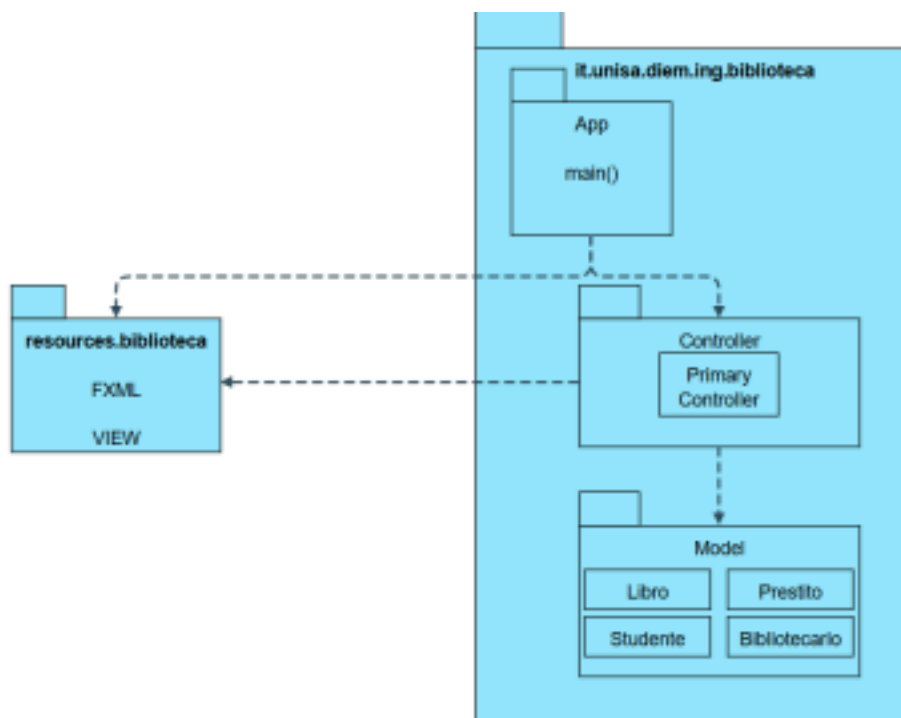
Controller: (it.unisa.diem.ing.biblioteca.controller)

- **Responsabilità:** Fa da tramite per il model e la view gestendo eventi generati dall'utente (click pulsante) e invoca i metodi delle classi nel model.

• **Classi principali:** Controller, ControllerDelete, ControllerHomepage, ControllerInfo, ControllerLibri, ControllerLogin, ControllerLogout, ControllerPopupLibro, ControllerPopupPrestiti, ControllerPopupStudenti, ControllerPrestiti, ControllerRitorno, ControllerStudenti.

- **Dipendenze:** Dipende dal Model (per accedere ai dati) e dalla View (per cambiare l'interfaccia utente).

Diagramma dei Package:



App -> View/Controller: Il modulo App è il punto di ingresso e dipende dagli altri perché deve caricare i file FXML.

Controller -> Model: I Controller hanno una dipendenza diretta verso i Model perché chiamano i metodi delle classi (es. Libro.ricercaLibro()) per eseguire le operazioni.

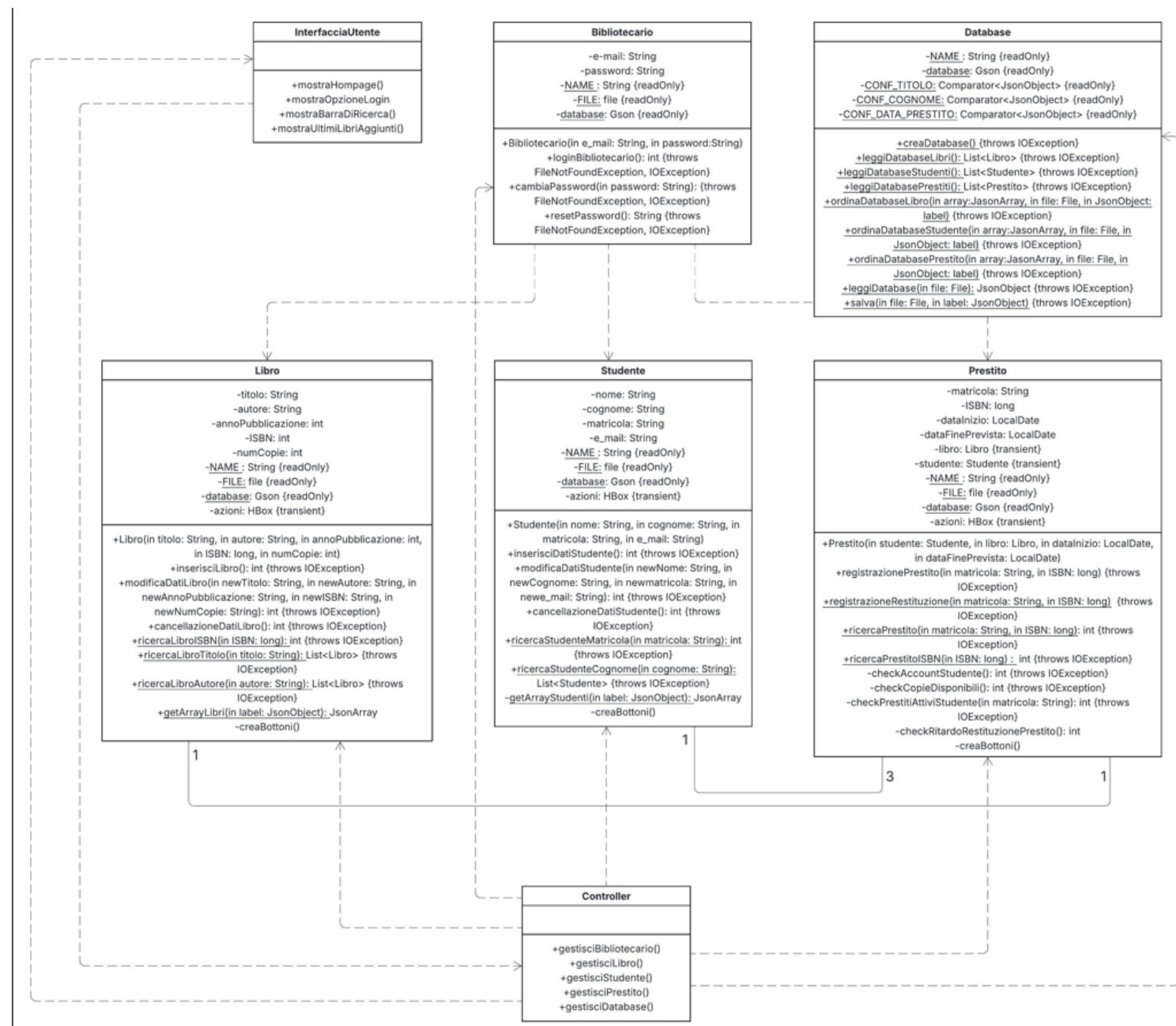
Controller -> View: I Controller sono collegati alle View (tramite le annotazioni @FXML) perché devono rispondere ai click dei pulsanti.

Model: È il pacchetto alla base della gerarchia e non ha frecce in uscita verso gli altri perché è

indipendente dall'interfaccia grafica.

2. Modello statico

Diagramma delle Classi:



Per leggibilità del diagramma delle classi, si è preferito omettere i getter ed i setter.

Descrizione delle Classi Principali:

Package biblioteca.model

• App

- **Responsabilità:** Gestisce lo Stage principale e il caricamento delle risorse. ◦

Metodi Pubblici: start().

• Libro

- **Responsabilità:** Gestisce la creazione, modifica, cancellazione, visualizzazione e ricerca legata al catalogo di libri
- **Attributi Principali:** titolo (String), autore (String), annoPubblicazione (int), ISBN (String), numCopie(int), NAME (String), FILE (File), database (Gson) .

■ **Metodi Pubblici:**

- getAzioni(): Prende la colonna vuota Azioni e la popola con i bottoni (in questo caso un bottone per la modifica e l'eliminazione).
- getTitolo(): Restituisce il titolo.
- getAutore(): Restituisce l'autore.
- getAnnoDiPubblicazione(): Restituisce l'anno di pubblicazione.
- getIsbn(): Restituisce l'ISBN.
- getNumCopie(): Restituisce il numero di copie del libro.
- inserisciLibro(): Crea e salva una nuovo libro.
- modificaDatiLibro (in newTitle: String , in newAuthor: String, in newAnnoPubblicazione: String , in newISBN: String, in newNumCopie: String): Aggiorna gli attributi di un libro esistente.
- cancellazioneDatiLibro(): Rimuove il libro dal database e catalogo.
- ricercaLibroISBN(in ISBN: Long): Restituisce i libri corrispondenti all'ISBN dato.
- ricercaLibroAutore(in autore: String): Restituisce i libri corrispondenti all'autore dato.
- ricercaLibroTitolo(in titolo: String): Restituisce i libri corrispondenti al titolo dato.
- **Relazioni Rilevanti:** È associato alla classe Prestito (un prestito riguarda un libro).

• Studente

- **Responsabilità:** Gestisce la creazione, modifica, cancellazione,

visualizzazione e ricerca dell'elenco degli studente

- **Attributi Principali:** nome (String), cognome (String), matricola (String), email (String), NAME (String), FILE (File), database (Gson) .
 - **Metodi Pubblici:**
 - `getAzioni()`: Prende la colonna vuota Azioni e la popola con i bottoni (in questo caso un bottone per: modifica, elimina ed info).
 - `getNome()`: Restituisce il nome.
 - `getCognome()`: Restituisce il cognome.
 - `getMatricola()`: Restituisce la matricola.
 - `getE_Mail()`: Restituisce l'email.
 - `inserisciDatiStudente()`: Registra un nuovo studente.
 - `modificaDatiStudente(in newNome: String, in newCognome: String , in newMatricola: String , in newE_mail: String)`: Aggiorna i dati anagrafici.
 - `cancellazioneDatiStudente()`: Rimuove lo studente.
 - `ricercaStudenteMatricola(in matricola: String)`: Restituisce l'indice corrispondente alla matricola data nella lista degli studenti.
 - `ricercaStudenteCognome(in cognome: String)`: Restituisce l'indice corrispondente al cognome dato nella lista degli studenti.
- **Relazioni Rilevanti:** È associato alla classe Prestito (uno studente effettua un prestito).

• Prestito

- **Responsabilità:** Classe che gestisce il prestito tra Studente e Libro.
- **Attributi Principali:** dataInizio (LocalDate), dataFinePrevista (LocalDate), studente (Studente), libro (Libro), matricola (String), ISBN (long), NAME (String), FILE (File), database (Gson) .
 - **Metodi Pubblici:**
 - `getAzioni()`: Prende la colonna vuota Azioni (in questo caso si chiama Restituzione) e la popola con i bottoni (in questo caso un solo bottone per la restituzione).
 - `getMatricola()`: Restituisce la matricola.
 - `getIsbn()`: Restituisce l'ISBN.
 - `getDataInizio()`: Restituisce la data di inizio del prestito.
 - `getDataFinePrevista()`: Restituisce la data di fine del prestito.
 - `registrazionePrestito(in matricola: String,in ISBN: long)`: Avvia un nuovo prestito e decrementa il numero di copie del libro.
 - `registrazioneRestituzione(in matricola: String,in ISBN: long)`: Chiude il prestito e incrementa il numero di copie del libro.
 - `ricercaPrestitoIsbn(in ISBN: Long)`: restituisce l'indice del prestito

trovato.

- checkAccountStudente(): Controlla che lo Studente sia registrato nel sistema.
- checkCopieDisponibili(in libro: Libro): Controlla che ci siano ancora copie disponibili del libro.
- checkPrestitiAttiviStudente(in matricola: String): Controlla che lo studente non abbia più di 3 prestiti attivi.
- checkRitardoRestituzionePrestito(): Controlla che lo studente non abbia prestiti attivi ma in ritardo di restituzione.
- **Relazioni Rilevanti:** Dipende da Libro e Studente.

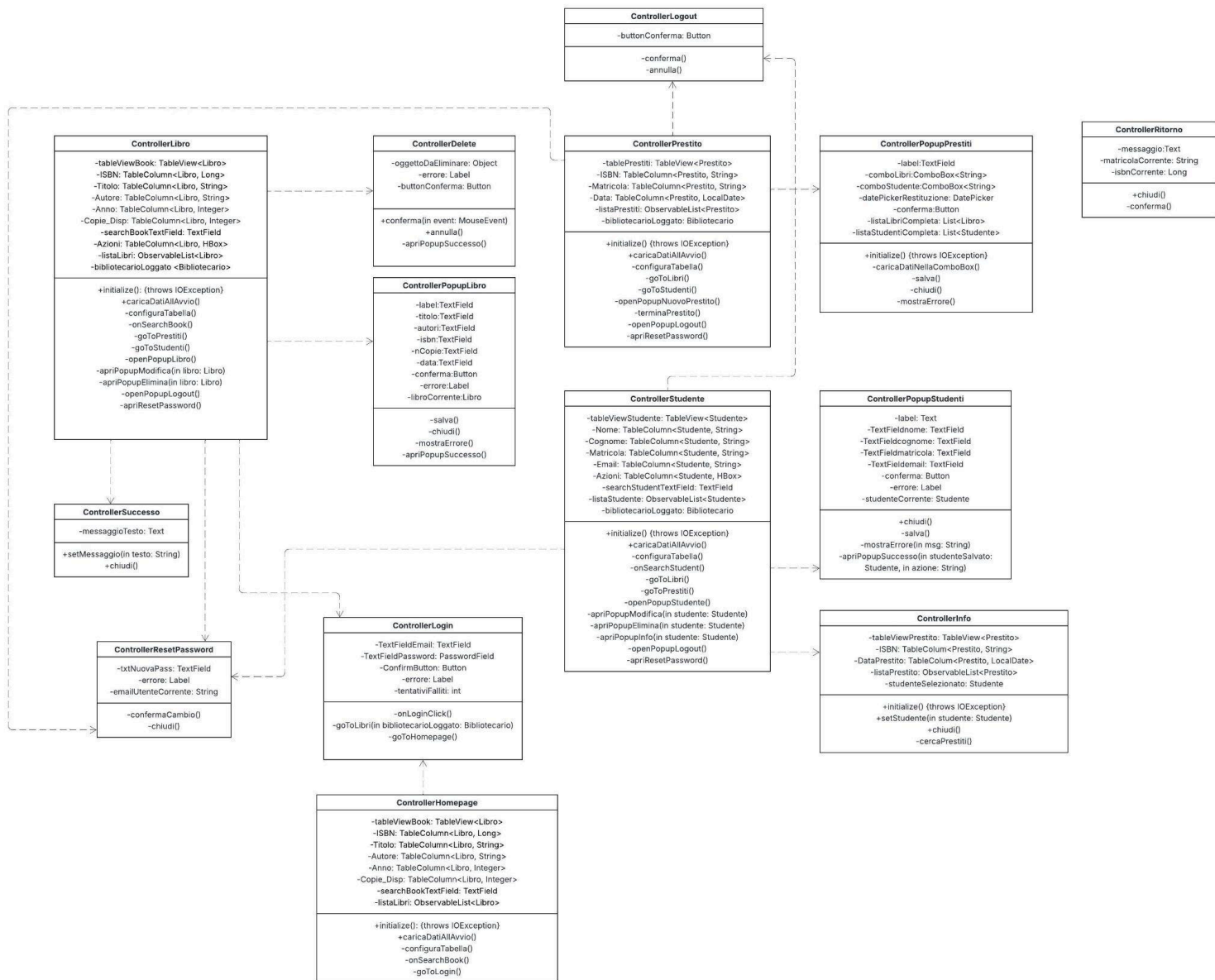
• Bibliotecario

- **Responsabilità:** Gestisce l'accesso alle funzionalità amministrative. ○

Attributi Principali: email (String), password (String), NAME (String), FILE (File), database (Gson) .

- **Metodi Pubblici:**
 - getEmail():
 - loginBibliotecario(): Verifica le credenziali per concedere l'accesso alla dashboard amministrativa.
 - cambiaPassword(): Permette al bibliotecario di cambiare la password.
 - resetPassword(): Permette al bibliotecario di resettare la password.

Package biblioteca.controller



Per leggibilità del diagramma delle classi, si è preferito omettere i getter ed i setter.

• ControllerLogin

- **Responsabilità:** Gestisce il processo di autenticazione del bibliotecario.
- **Attributi Principali:** TextFieldEmail (TextField), TextFieldPassword (PasswordField), ConfirmButton (Button), errore (Label), tentativiFalliti (int).
- **Metodi Pubblici:** **initialize():** Inizializza la tabella e carica i dati all'avvio.
- **Relazioni Rilevanti:** Interagisce con la classe Bibliotecario per la verifica delle credenziali.

• ControllerHomepage

- **Responsabilità:** Gestisce la visualizzazione pubblica del catalogo libri per utenti non loggati.
- **Attributi Principali:** tableViewBook (TableView<Libro>), ISBN (TableColumn<Libro, Long>), Titolo (TableColumn<Libro, String>), Autore (TableColumn<Libro, String>), Anno (TableColumn<Libro, Integer>), Copie_Dis (TableColumn<Libro, Integer>), searchBookTextField (TextField), listaLibri (ObservableList<Libro>).
- **Metodi Pubblici:**
 - initialize(): Inizializza la tabella e carica i dati all'avvio.
 - caricaDatiAllAvvio(): *

• ControllerResetPassword

- **Responsabilità:** Gestisce la procedura per reimpostare la password del bibliotecario.
- **Attributi Principali:** txtNuovaPass (TextField), errore (Label), emailUtenteCorrente (String).
- **Metodi Pubblici:** N/A.

• ControllerLibri

- **Responsabilità:** Gestisce la schermata principale per l'amministrazione dei libri (visualizzazione, ricerca, e accesso alle funzioni di modifica).
- **Attributi Principali:** tableViewBook (TableView<Libro>), ISBN(TableColumn<Libro, Long>), Titolo(TableColumn<Libro, String>), Autore(TableColumn<Libro, String>), Anno(TableColumn<Libro, Integer>), Copie_Dis(TableColumn<Libro, Integer>), searchBookTextField(TextField), Azioni(TableColumn<Libro, HBox>), listaLibri(ObservableList<Libro>), bibliotecarioLoggato (Bibliotecario).
- **Metodi Pubblici:**
 - initialize() e configuraTabella(): Preparano la TableView.
 - caricaDatiAllAvvio(): Legge il database e popola coi libri la tabella, o da errore se non ne trova. Viene anche usata per aggiornare la tabella ad ogni modifica.
- **Relazioni Rilevanti:** Gestisce istanze di Libro e naviga verso ControllerPopupLibro e ControllerDelete.

• ControllerStudente

- **Responsabilità:** Gestisce la schermata principale per l'amministrazione degli studenti iscritti.
- **Attributi Principali:** tableViewStudenti (TableView<Studente>), nome(TableColumn<Studente, String>), cognome(TableColumn<Studente, String>), Matricola(TableColumn<Studente, String>), Email(TableColumn<Studente, String>), searchStudentTextField(TextField), Azioni(TableColumn<Studente, HBox>), listaStudente(ObservableList<Studente>), bibliotecarioLoggato (Bibliotecario).

- **Metodi Pubblici:**

- initialize(): Carica la lista studenti.
- caricaDatiAllAvvio(): Legge il database e popola con gli studenti a tabella, o da errore se non ne trova. Viene anche usata per aggiornare la tabella ad ogni modifica.

- **Relazioni Rilevanti:** Gestisce istanze di Studente e naviga verso ControllerPopupStudenti e ControllerInfo.

- **ControllerPrestito**

- **Responsabilità:** Gestisce la visualizzazione e le operazioni sui prestiti attivi.
- **Attributi Principali:** tablePrestiti (TableView<Prestito>), ISBN (TableColumn<Prestito, String>), Matricola (TableColumn<Prestito, String>), Data (TableColumn<Prestito, LocalDate>), listaPrestiti (ObservableList<Prestito>), bibliotecarioLoggato (Bibliotecario).
- **Metodi Pubblici:**
 - initialize(): Usa la funzione creaDatabase e caricaDatiAllAvvio().
 - caricaDatiAllAvvio(): Legge il database e popola coi prestiti la tabella, o da errore se non ne trova. Viene anche usata per aggiornare la tabella ad ogni modifica.
- **Relazioni Rilevanti:** Gestisce istanze di Prestito e naviga verso ControllerPopupPrestiti.

- **ControllerPopupLibro**

- **Responsabilità:** Gestisce il form per l'aggiunta o la modifica dei dati di un libro.
- **Attributi Principali:** label (TextField), titolo (TextField), autori (TextField), isbn (TextField), nCopie (TextField), data (TextField), conferma (Button), errore (Label), libroCorrente (Libro).
- **Metodi Pubblici:**
 - setLibroDaModificare(in libro: Libro): Fa il setting di libroCorrente. Se questo non è null, dà la possibilità al bibliotecario di modificare il libro.

- **ControllerPopupStudenti**

- **Responsabilità:** Gestisce il form per l'aggiunta o modifica dati studente.
- **Attributi Principali:**

label (Text), TextFieldnome (TextField), TextFieldcognome (TextField), TextFieldmatricola (TextField), TextFieldemail (TextField), conferma (Button), errore (Label), studenteCorrente (Studente). □

- **Metodi Pubblici:**

□setStudenteDaModificare(in studente: Studente): Fa il setting di studenteCorrente. Se questo non è null, dà la possibilità al bibliotecario di modificare i dati dello studente.

□chiudi(): Cambia scena.

- **ControllerPopupPrestiti**

- **Responsabilità:** Gestisce il form per la creazione di un nuovo prestito.
- **Attributi Principali:** label:TextField, comboLibri (ComboBox<String>), comboStudente (ComboBox<String>), datePickerRestituzione (DatePicker), conferma (Button), listaLibriCompleta (List<Libro>), listaStudentiCompleta (List<Studente>).
- **Metodi Pubblici:**
 - initialize(): Usa la funzione creaDatabase() per accertarsi dell'esistenza del database, altrimenti lo crea. Imposta il titolo del popup prestito.

- **ControllerDelete**

- **Responsabilità:** Gestisce una finestra di conferma generica per l'eliminazione di un oggetto (Libro o Studente).
- **Attributi Principali:** oggettoDaEliminare (Object), errore (Label), buttonConferma (Button).
- **Metodi Pubblici:**
 - conferma(in event: MouseEvent): Esegue l'eliminazione effettiva dal sistema.
 - annulla(): Chiude la finestra senza eliminare.

- **ControllerSuccesso**

- **Responsabilità:** Mostra un messaggio di feedback positivo all'utente.
- **Attributi Principali:** messaggioTesto(Text).
- **Metodi Pubblici:**
 - setMessaggio(in testo: String): Fa il setting di messaggio.
 - chiudi(): Cambia scena.

- **ControllerInfo**

- **Responsabilità:** Visualizza dettagli specifici, probabilmente lo storico dei prestiti di un singolo

studente selezionato.

- **Attributi Principali:** tableViewPrestito (TableView<Prestito>), ISBN (TableColumn<Prestito, String>), DataPrestito (TableColumn<Prestito, LocalDate>), listaPrestito (ObservableList<Prestito>), studenteSelezionato (Studente).
- **Metodi Pubblici:**
 - initialize(): Usa la funzione creaDatabase() per accertarsi dell'esistenza del database, altrimenti lo crea.
 - setStudente(in studente: Studente): Fa il setting di studente e chiama il metodo cercaPrestiti.

• ControllerLogout

- **Responsabilità:** Gestisce la conferma di disconnessione del bibliotecario.
- **Attributi Principali:** buttonConferma(Button).
- **Metodi Pubblici:** N/A.

• ControllerRitorno

- **Responsabilità:** Sembra gestire la conferma o i dettagli relativi alla restituzione di un libro (Ritorno del prestito).
- **Attributi Principali:** messaggio (Text), matricolaCorrente (String), isbnCorrente (Long)
- **Metodi Pubblici:**
 - chiudi(): Cambia la scena.
 - setDatiRestituzione(in matricola: String, in isbn: Long): Lo studente con la matricola ricevuta restituisce il libro con l'ISBN ricevuto in data odierna.

Scelte Progettuali:

- **Coesione:** Ogni classe ha una singola responsabilità ben definita. Le classi del Model si occupano solo dei dati, i Controller solo della gestione delle azioni e le View solo dell'interfaccia grafica. Questo rispetta il **Single Responsibility Principle (SRP)**.
- **Accoppiamento:** L'uso del pattern MVC riduce le dipendenze perché le classi del Model funzionano a discapito di come è implementata l'interfaccia.
- **Incapsulamento:** Tutti gli attributi delle classi sono private e accessibili solo tramite metodi pubblici controllati.

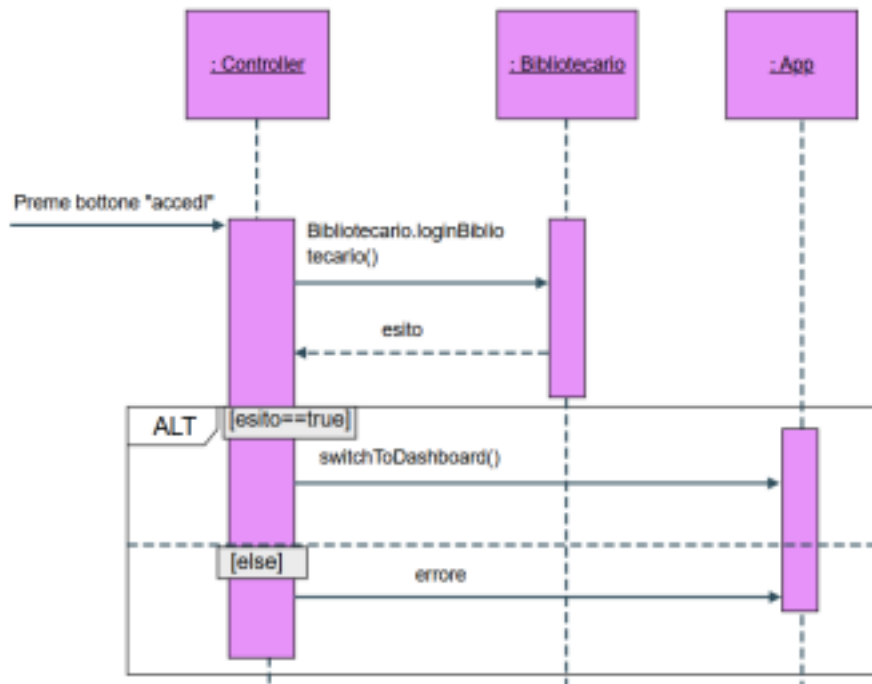
3. Modello dinamico

Descrizione delle Interazioni:

IF-01: Login (Bibliotecario)

- **Descrizione:** L'utente accede alle funzionalità amministrative.

• Diagramma di Sequenza

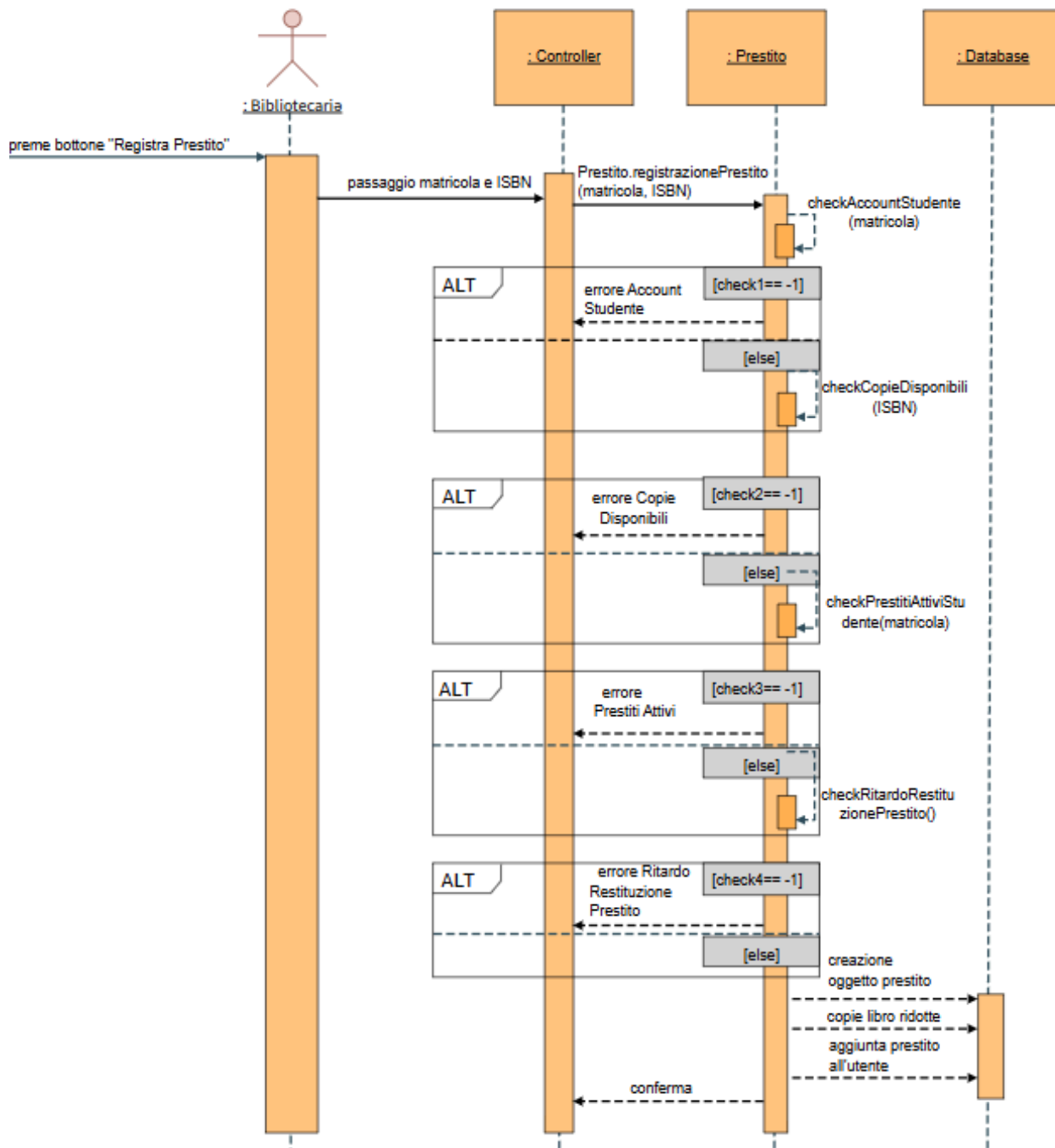


• Commento al diagramma:

1. L'utente inserisce le credenziali nella schermata di login e preme "Accedi".
2. Il Controller recupera i dati dai campi di testo.
3. Il controller invoca il metodo `Bibliotecario.loginBibliotecario()`.
4. Il metodo `Bibliotecario.loginBibliotecario()` verifica le credenziali e restituisce un `esito` (booleano).
5. Se positivo, il controller chiama `switchToDashboard()` (in App?) per mostrare la Dashboard. Altrimenti, mostra un errore.

BF-11: Registrazione prestito

- **Descrizione:** Assegnazione di un libro a uno studente.
- **Diagramma di sequenza:**



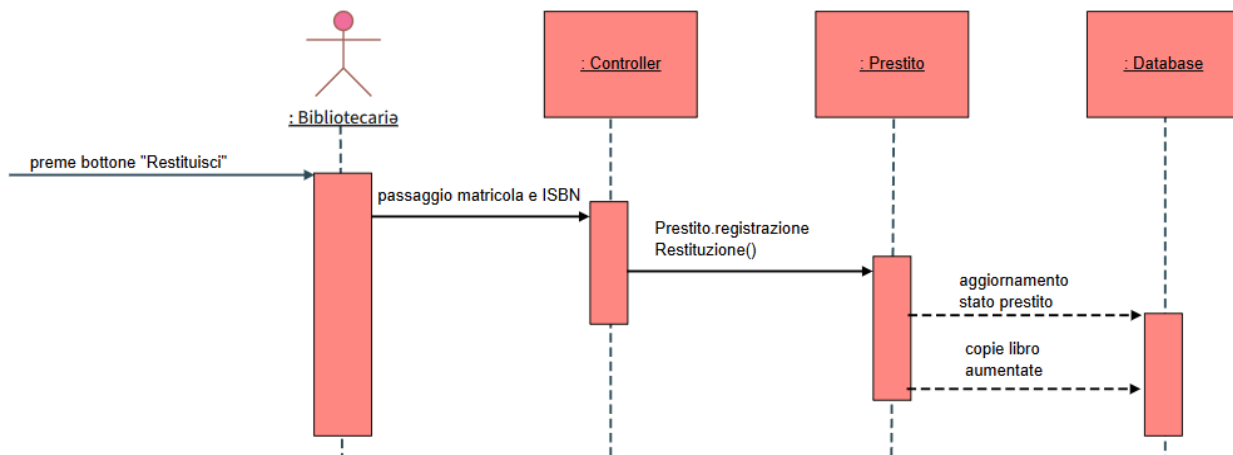
Commento al diagramma:

1. Il Bibliotecario seleziona uno Studente e un Libro dalla sezione per registrare un prestito e clicca "Registra Prestito"
2. Il Controller intercetta l'evento e chiama `Prestito.registrazionePrestito()`.
3. `Prestito` chiama `checkAccountStu-dente`: verifica se lo studente è presente nel database.
4. `Prestito` chiama `checkCopieDisponibili()`: verifica se le copie sono > 0 .
5. `Prestito` chiama `checkPrestitiAttiviStu-dente()`: verifica se lo studente ha 3 o più prestiti attivi.

6. Prestito chiama checkRitardoRestituzionePrestito(): verifica se lo studente ha un prestito in ritardo.
7. Viene creato un oggetto Prestito.
8. Libro decrementa le copie disponibili.
9. Il nuovo prestito viene salvato nel database.

BF-13: Registrazione restituzione

- **Descrizione:** Chiusura di un prestito e rientro del libro.
- **Diagramma di sequenza:**



Commento al diagramma:

1. Il bibliotecario seleziona un prestito attivo e clicca "Restituisci".
2. Il controller invoca **Prestito.registrazioneRestituzione()**
3. Il metodo rimuove il prestito dal database e incrementa le copie disponibili del libro associato.

4. Design dell'interfaccia utente

Mock-up delle Schermate Principali:

- Catalogo libri visto dagli ospiti

[Login](#)

Catalogo Libri

Titolo	Autore	ISBN	Anno	Num. Copie
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0679608552	2014	3

1-10 of 97

Rows per page: 10

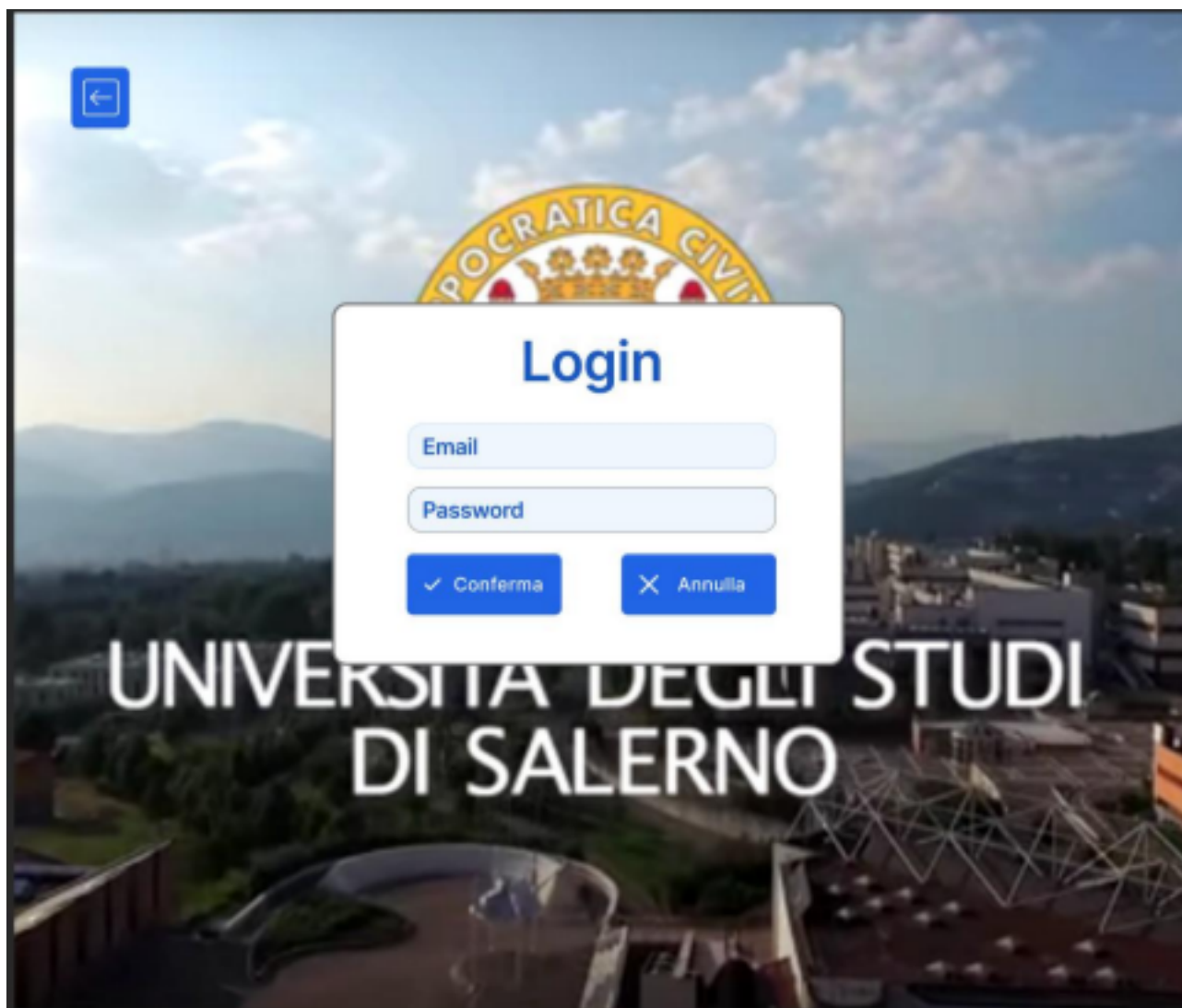
1/10

In alto è presente una barra di ricerca per il catalogo dei libri, a destra un pulsante per effettuare il login ed avere accesso alla dashboard amministrativa.

Al centro una tabella che mostra: Titolo, Autore, ISBN e num di Copie disponibili.

In basso è presente una barra per modificare il numero di libri visti nella pagina e per passare alla pagina successiva.


• Schermata di Login



Al centro della finestra sono presenti, due campi di testo ("Email" e "Password") e un pulsante per confermare o cancellare il form di login. Ma inserendo dati errati, apparirà una label rossa sotto il pulsante con scritto "Credenziali non valide".

In alto a destra una freccia per tornare indietro alla homepage del catalogo libri per gli ospiti.

- **Catalogo Libri - dashboard amministrativa**



Cerca Libri

Libri





















Prestiti

Studenti

Catalogo Libri

+

Aggiungi libro

Titolo	Autore	ISBN	Anno	Num. Copie	
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 
Java: The Complete Reference (5th ed.)	Herbert Schildt	978-0071806552	2014	3	 

1-10 of 97

Rows per page: 10

1/10

Come per quella degli ospiti è presente una barra di ricerca, la tabella con i libri e la barra per avanzare nelle pagine successive. Nella schermata è però anche presente un nuovo bottone situato sopra la tabella che permette di aggiungere nuovi libri e per ogni libro sono presenti due icone intuitive alla fine di ogni riga: una matita che permette la modifica dei dati del libro e un cestino che permette la cancellazione del libro dal sistema. È inoltre presente un tasto per l'ordinamento dei libri per titolo.

Inoltre, è presente un menu a sinistra della schermata che permette di navigare la dashboard amministrativa e tutte le sue funzionalità: il primo link permette di passare al catalogo di libri (quella attuale), il secondo permette la visione dei prestiti ancora attivi e il terzo di navigare gli studenti iscritti al sistema.

• Prestiti Attivi – dashboard amministrativa

Prestiti Attivi [+ Nuovo Prestito](#)

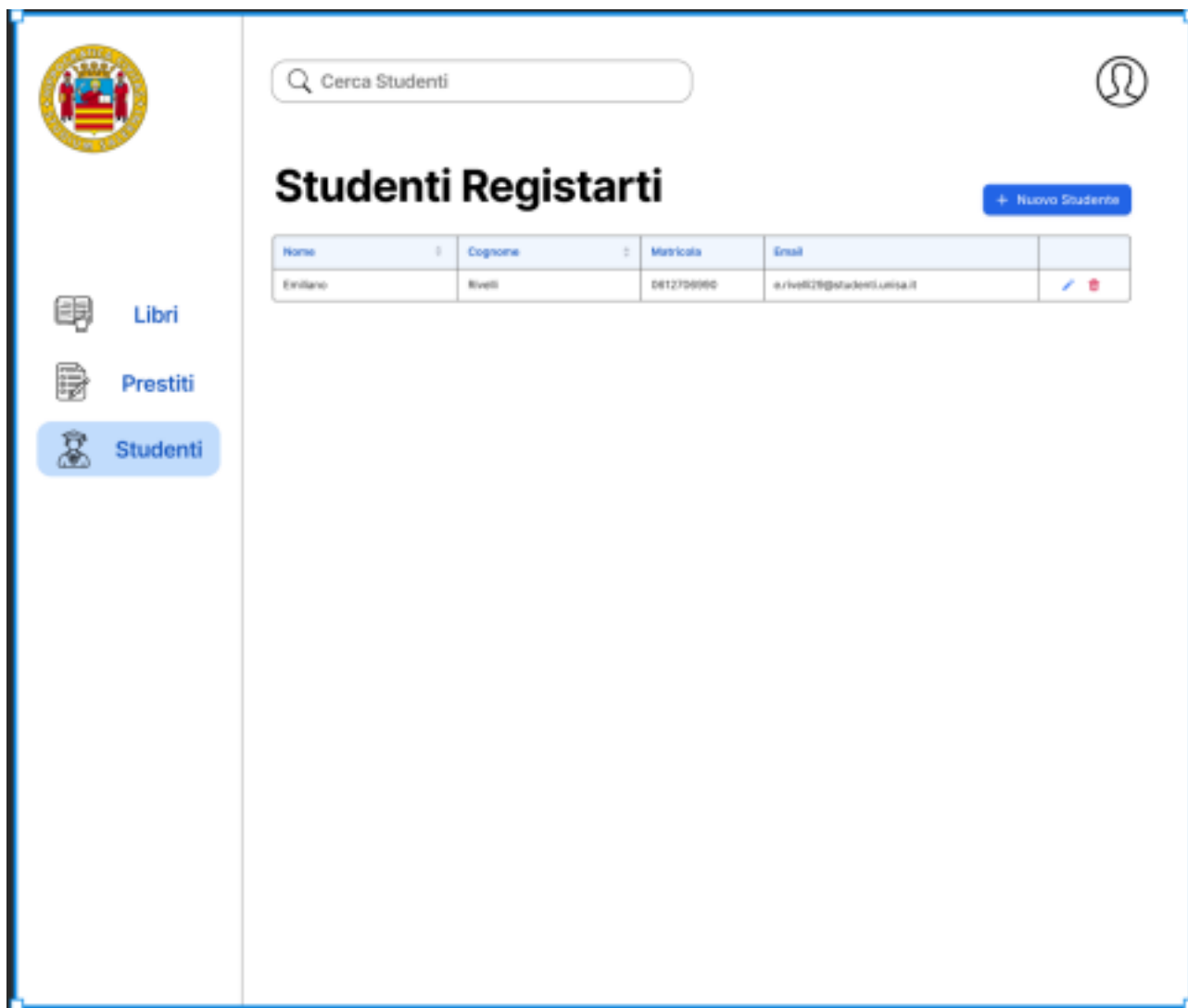
ISBN - Libro	Titolo - Libro	Matricola - Studente	Data Restituzione	
978-9071800552	Java: The Complete Reference (9th ed.)	0612706990	07/12/25	<input checked="" type="checkbox"/> Restituito
978-9071800552	Java: The Complete Reference (9th ed.)	0612706990	30/12/25	<input checked="" type="checkbox"/> Restituito

La stessa schermata precedente con barra di ricerca (cerca i prestiti), menu a sinistra e tabella centrale che questa volta contiene l'elenco dei prestiti attivi: ISBN, Titolo del libro, Matricola, Data di restituzione e un bottone sopra che permette di inserire un nuovo prestito.

I prestiti con data di restituzione antecedente a quella attuale sono evidenziati con una scritta rossa.

A destra della tabella è presente per ogni prestito un bottone per chiuderlo indicando l'avvenuta restituzione; e accanto alla scritta di data è presente un'icona che permette l'ordinamento della tabella per data di restituzione.

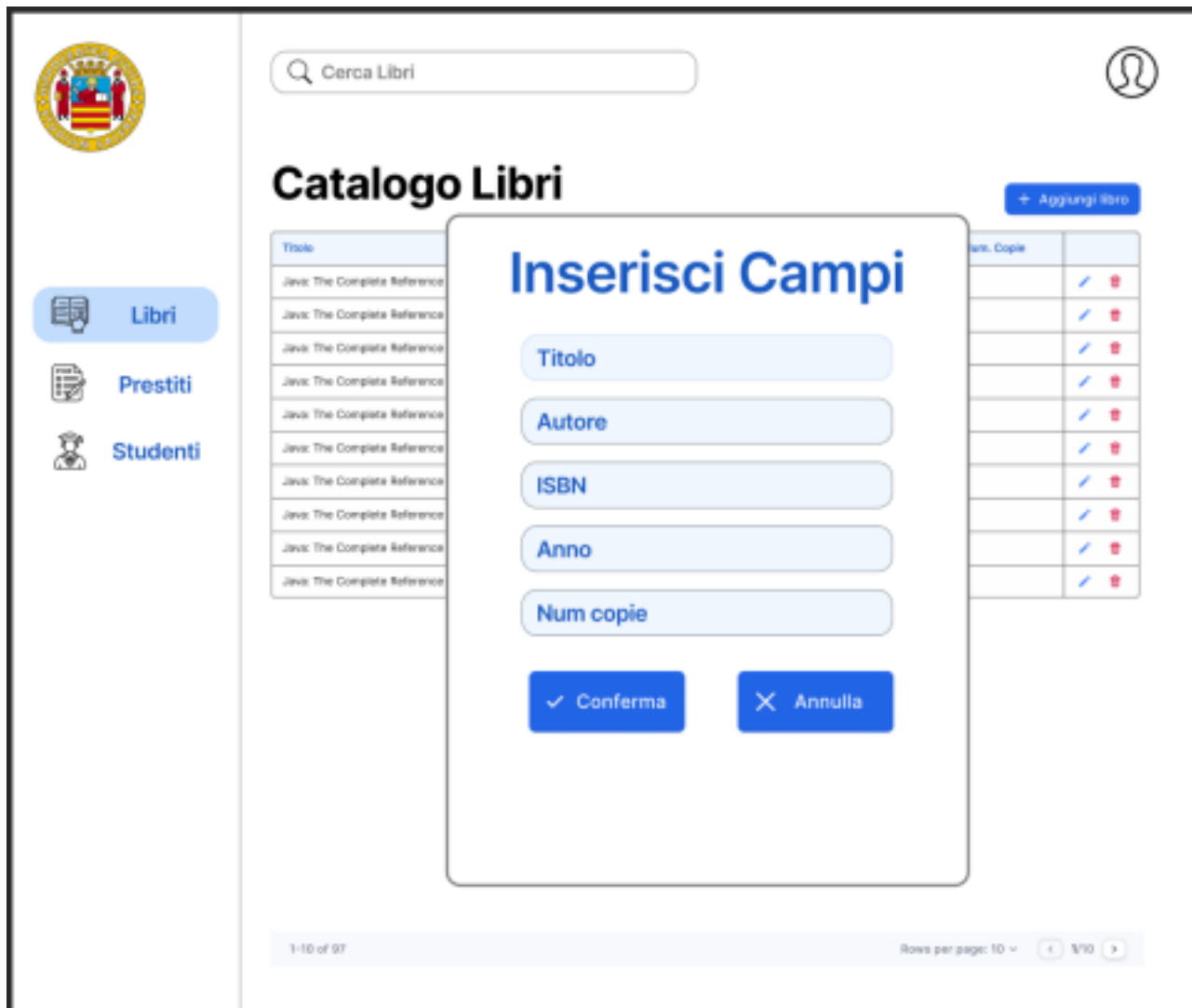
- **Studenti Registrati – dashboard amministrativa**



La stessa schermata precedente con barra di ricerca (cerca gli studenti), menu a sinistra e tabella centrale che questa volta contiene l'elenco degli studenti registrati: Nome, Cognome, Matricola, Email e un bottone sopra che permette di inserire un nuovo studente.


Per ogni studente sono presenti due icone intuitive alla fine di ogni riga: una matita che permette la modifica dei dati dello studente e un cestino che permette la cancellazione dello studente dal sistema. È inoltre presente un tasto per l'ordinamento degli studenti per nome o cognome.

- **Popup per Inserimento o modifica di libri e studenti**




Finestra popup che si apre sopra la dashboard. Contiene form con validazione (es. non permette di salvare se l'ISBN è vuoto) con tasti "Conferma" e "Annulla".

- **Popup per cancellazione di libri e studenti**



Cerca Libri



Catalogo Libri





























Libri

Prestiti

Studenti

+

Aggiungi libro

Titolo	Autore	ISBN	Anno	Num. Copie	
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 
Java: The Complete Reference (9th ed.)	Herbert Schildt	978-0371839552	2014	3	 

Sei sicuro di voler eliminare questo campo?

✓ Conferma

✗ Annulla

1-10 of 97

Rows per page: 10

1

2

3

4

5

6

7

8

9

10

Finestra popup che si apre sopra la dashboard. Contiene una conferma della voluta cancellazione del libro con tasti "Conferma" e "Annulla".

- Pop up per inserimento di un nuovo prestito

The screenshot shows a web application interface for a library. On the left is a sidebar with a logo and navigation links: 'Libri', 'Prestiti' (highlighted), and 'Studenti'. The main area has a search bar 'Cerca Prestiti' and a user profile icon. Below the search bar is the section 'Prestiti Attivi'. A table lists active loans with columns 'ISBN - Libro' and 'Titolo'. A modal window titled 'Inserisci Campi' is open, containing two sections: 'ISBN' and 'Matricola'. Each section has a search input and a list of suggestions. At the bottom of the modal are 'Conferma' and 'Annulla' buttons. To the right of the modal, a small table shows loan status with 'Restituito' buttons.

ISBN - Libro	Titolo
978-0071808552	Java: T
978-0071808552	Java: T

Inserisci Campi

ISBN

978-0071808552

978-0071808554

978-0071808587

978-0071809875

978-0078764523

Matricola

0612706990

0612706996

06127069356

06127065734

06127069438

✓ Conferma

✗ Annulla

✓ Restituito
✓ Restituito

Finestra popup che si apre sopra la dashboard. Contiene un form con validazione (es. non permette di salvare se l'ISBN è vuoto o non esiste come libro nel sistema) con menu a scorrimento automatico per l'inserimento facilitato del isbn e matricola.

- **Popup per la restituzione di un prestito**



Finestra popup che si apre sopra la dashboard. Contiene una conferma per l'effettiva restituzione del libro giusto da parte dello studente corretto.