

UNIVERSITÀ POLITECNICA DELLE MARCHE

INGEGNERIA DEL SOFTWARE

Software per la gestione della Biblioteca Universitaria

Autori:

Francesco FINIZII
Kristian LIKAJ



17 ottobre 2023

Indice

1 DESCRIZIONE	4
1.1 Accesso alla piattaforma	4
1.2 Catalogo Biblioteca	4
1.3 Prestito librario	5
1.4 Prenotazione posto aula studio	5
1.5 Glossario di progetto	7
2 ANALISI DEI REQUISITI	8
2.1 Requisiti funzionali	9
2.1.1 Gestione Catalogo	9
2.1.2 Gestione Prestiti	9
2.1.3 Gestione Prenotazioni	10
2.1.4 Gestione Profilo	10
2.1.5 Gestione Utente	10
2.1.6 Gestione Sistema	10
2.2 Requisiti non funzionali	11
2.2.1 Generali	11
2.2.2 Implementazione	12
3 CASI D'USO	13
3.1 Gestione Catalogo	15
3.1.1 AggiungiDocumento	17
3.1.2 ConsultazioneCatalogo	18
3.1.3 VisualizzaDocumento	19
3.1.4 ModificaDocumento	20
3.1.5 RimuoviDocumento	21
3.2 Gestione Prenotazioni	22
3.2.1 AggiungiPrenotazione	23

3.2.2	VisualizzaPrenotazione	24
3.2.3	RimuoviPrenotazione	25
3.2.4	VisualizzaListaPrenotazioni	26
3.2.5	RicercaTurniDisponibili	27
3.3	Gestione Sistema	28
3.3.1	Autenticazione	30
3.3.2	Backup	31
3.3.3	SospensioneAccount	32
3.3.4	TerminaPrenotazioni	33
3.4	Gestione Prestiti	34
3.4.1	EffettuaPrestito	36
3.4.2	VisualizzaPrestito	37
3.4.3	ProrogaPrestito	38
3.4.4	TerminaPrestito	39
3.4.5	AttivazioneAccount	40
3.4.6	VisualizzaListaPrestiti	41
3.5	Gestione Utente	42
3.5.1	VisualizzaListaUtenti	43
3.5.2	VisualizzaUtente	44
3.6	Gestione Profilo	45
3.6.1	VisualizzaAccount	47
3.6.2	ModificaAccount	48
3.6.3	EliminaAccount	49
3.6.4	RegistrazioneAccount	50
4	MATRICE DI MAPPING	51
5	CLASSI DI ANALISI	53
6	CLASSI DI PROGETTAZIONE	55
6.1	Model	56
6.2	Controller	57
6.3	View	58
6.3.1	Generale	59
6.3.2	Studente	64
6.3.3	Amministratore	74

7 DIAGRAMMI DI SEQUENZA	93
7.1 AggiungiDocumento	94
7.2 ModificaDocumento	95
7.3 RimuoviDocumento	96
7.4 Autenticazione	97
7.5 EffettuaPrestito	98
7.6 VisualizzaListaPrenotazioni	99
7.7 VisualizzaPrestito	100
8 DIAGRAMMI DI ATTIVITÀ	101
8.1 Autenticazione	102
8.2 VisualizzaListaPrenotazioni	103
8.3 act-RicercaPrenotazione	104
8.4 TerminaPrestito	105
9 DIAGRAMMI DELLE MACCHINE A STATI	106
9.1 Prenotazione	107
9.2 Prestito	108
9.3 Studente	109
10 DIAGRAMMA DEI COMPONENTI	110
11 DIAGRAMMA DI DEPLOYMENT	111
12 UNIT TEST	112
12.1 test_studente	113
12.2 test_amministratore	113
12.3 test_libro	114
12.4 test_rivista	115
12.5 test_prestito	116
12.6 test_prenotazione	117
13 NOTE	118

1. DESCRIZIONE

Il progetto proposto consiste nella realizzazione di un software per la gestione del Sistema Bibliotecario dell'Università Politecnica delle Marche. Il software in questione si occupa di due principali servizi:

1. Prestito librario.
2. Prenotazione posto aula studio.

1.1 Accesso alla piattaforma

Per poter usufruire dei servizi offerti dal software, gli studenti e il personale della biblioteca si possono registrare gratuitamente utilizzando la matricola fornita dall'Ateneo. Come da regolamento, per completare la registrazione, il personale della biblioteca deve fornire il codice di sicurezza fornito dal Centro di Documentazione di Ateneo (CAD). Una volta avuto accesso alla piattaforma è possibile visualizzare, modificare o eliminare il proprio profilo utente. La rimozione del profilo è concessa solamente agli studenti che non hanno prestiti in corso. Il personale addetto alla gestione della piattaforma può ricercare un utente presente nel sistema e visualizzarne i dettagli e l'attuale stato.

1.2 Catalogo Biblioteca

Il Sistema Bibliotecario di Ateneo propone a tutti gli studenti un ricco patrimonio bibliografico. La piattaforma consente di consultare solamente l'archivio della Biblioteca Tecnico-Scientifica-Biomedica. Nel catalogo sono inclusi solamente libri e riviste in quanto unici documenti che possono essere concessi in prestito. Solo il personale addetto alla gestione della piattaforma può aggiungere, modificare ed eliminare un documento dal catalogo. A ogni utente è concesso verificare se un'opera è presente

nelle raccolte del Sistema bibliotecario e vederne la disponibilità e i dettagli. Inoltre la piattaforma permette di filtrare i risultati e visualizzarli nell'ordine desiderato.

1.3 Prestito librario

Il prestito librario è concesso agli studenti dell'Ateneo per non più di 2 volumi contemporaneamente. Il prestito è personale e non può essere trasferito ad altri. La durata del prestito è di 30 giorni. È possibile prorogare tale scadenza per altri 7 giorni. Gli studenti iscritti possono fare richieste di prestito tramite la piattaforma. La richiesta viene presa in carico dal sistema e dal successivo giorno feriale è possibile recarsi in sede per ritirare il documento fornendo il codice identificativo del prestito al personale della biblioteca. Una volta restituito il documento il personale della biblioteca provvede a terminare il prestito dello studente tramite piattaforma. I prestiti devono essere restituiti entro i tempi previsti. Lo studente che non risulta in regola con le restituzioni viene disabilitato automaticamente dal servizio e non può effettuare nuovi prestiti, prenotazioni o proroghe. La riabilitazione al servizio di prestito avviene automaticamente con la restituzione del documento o, in caso di smarrimento e deterioramento, con l'acquisto di titoli sostitutivi che verranno indicati dal personale.

1.4 Prenotazione posto aula studio

Le biblioteche del CAD sono aperte per i servizi in presenza dal lunedì al venerdì dalle 8:30 alle 19:00. Tramite piattaforma gli studenti iscritti possono prenotare il proprio posto in una delle aule dell'Ateneo. Le aule studio sono a disposizione nelle seguenti sedi:

- Biblioteca Tecnico-Scientifica-Biomedica (Montedago)
- Biblioteca Economico-Giuridica-Sociologica (Villarey)
- Biblioteca Medica (Torrette)

Ogni studente può effettuare massimo 1 prenotazione al giorno distante massimo 3 giorni feriali dal giorno corrente. Ogni prenotazione può durare massimo 2 ore e non è possibile selezionare diverse fasce orarie nell'arco della stessa giornata. Il posto non può essere scelto, ma viene assegnato automaticamente dal software il quale lo associa alla matricola dello studente in questione. Per consentire la massima disponibilità

delle postazioni, gli studenti sono tenuti a cancellare in anticipo le prenotazioni nel caso in cui non dovessero essere presenti. La cancellazione avviene esclusivamente tramite piattaforma.

1.5 Glossario di progetto

PAROLA	DEFINIZIONE	SINONIMI	OMONIMI
Autenticazione	Riconoscimento dell'Utente tramite matricola e Password.	Accesso	Nessuno
Registrazione	Creazione account da parte dell'utente per accedere al software.	Nessuno	Nessuno
Utente	Chi utilizza il software.	Nessuno	Nessuno
Studente	Persona iscritta all'Ateneo che utilizza i servizi offerti dal software.	Nessuno	Nessuno
Amministratore	Impiegato addetto all'amministrazione della biblioteca.	Nessuno	Nessuno
Bibliotecario/a	Addetto/a al servizio della biblioteca.	Nessuno	Nessuno
Assistente	Addetto che esegue i compiti affidati dal bibliotecario	Nessuno	Nessuno
Archivista	Addetto specializzato nella gestione dell'archivio bibliografico	Nessuno	Nessuno
Matricola	Codice identificativo fornito dall'Ateneo a studenti e personale universitario.	Nessuno	Nessuno
Password	Serie di caratteri alfanumerici che costituisce la parola d'ordine per accedere al sistema.	Nessuno	Nessuno
Ruolo	Compito svolto dal personale della biblioteca.	Nessuno	Nessuno
Codice sicurezza	Codice fornito dal CAD al personale addetto alla gestione del software.	Nessuno	Nessuno
Prestito	Cessione temporanea di un documento a uno studente da parte della biblioteca.	Nessuno	Nessuno
Prenotazione	Impegno a occupare e diritto ad aver riservato un posto studio in apposite strutture offerte dall'Ateneo.	Nessuno	Nessuno
Catalogo	Archivio ordinato e sistematico dei documenti prestabili dalla biblioteca.	Nessuno	Nessuno
Sospensione	Interruzione temporanea dell'accesso al software per lo studente.	Nessuno	Nessuno
Attivazione	Riabilitazione all'accesso al software per lo studente.	Nessuno	Nessuno
Turno	Periodo di tempo assegnato alla prenotazione effettuata tramite software.	Fascia oraria	Nessuno
Backup	Copia di sicurezza di tutti i dati archiviati nel sistema.	Nessuno	Nessuno

2. ANALISI DEI REQUISITI

I requisiti del sistema sono suddivisi in *requisiti funzionali* e *requisiti non funzionali* come illustrato in figura 2.1:

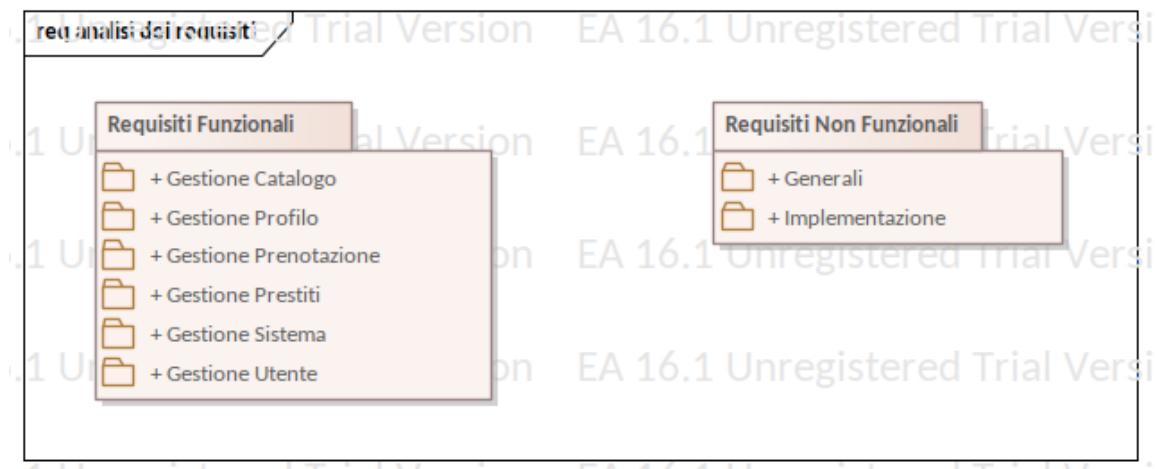


Figura 2.1: Analisi dei requisiti

2.1 Requisiti funzionali

I requisiti funzionali descrivono il comportamento del software. Sono stati suddivisi in 6 macro-categorie ognuna rappresentata da un package contenente i corrispondenti requisiti come mostrato in figura 2.3

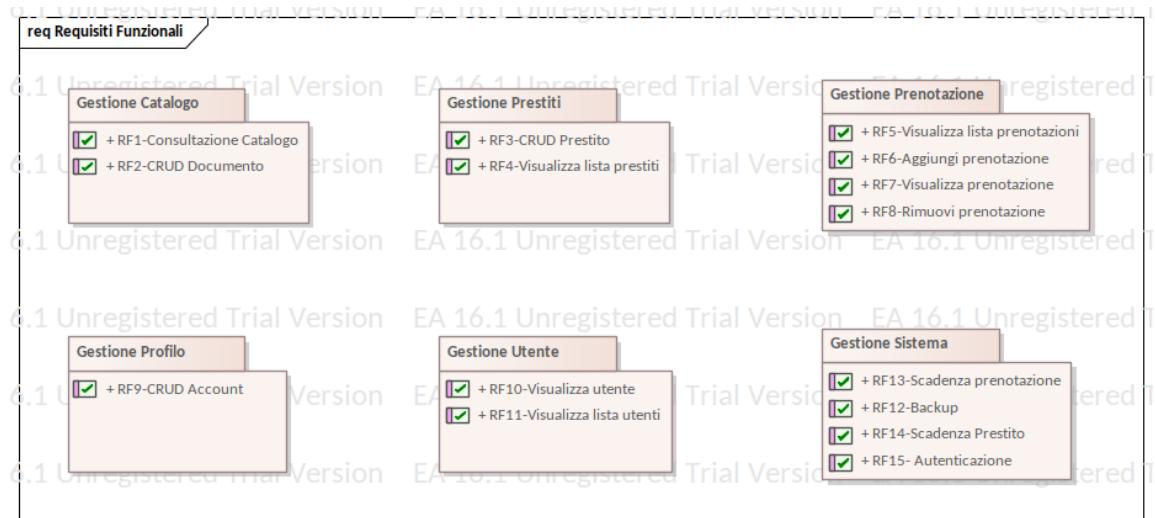


Figura 2.2: Requisiti Funzionali

2.1.1 Gestione Catalogo

- **RF1-Consultazione Catalogo:** Il sistema dovrà garantire la consultazione dei titoli disponibili al prestito librario e dei titoli presenti nell'archivio della biblioteca.
- **RF2-CRUD Documento:** Il sistema dovrà gestire le attività CRUD sui documenti.

2.1.2 Gestione Prestiti

- **RF3-CRUD Prestito:** Il sistema dovrà gestire le attività CRUD sui prestiti.
- **RF4-Visualizza lista prestiti:** Il sistema dovrà gestire la visualizzazione dei prestiti attualmente in corso.

2.1.3 Gestione Prenotazioni

- **RF5-Visualizza lista prenotazioni:** Il sistema dovrà gestire la visualizzazione delle prenotazioni presenti nelle relative sedi dell'Ateneo.
- **RF6-Aggiungi prenotazione:** Il sistema dovrà permettere agli studenti di prenotare un postazione finalizzata allo studio nelle apposite sedi dell'Ateneo.
- **RF7-Visualizza prenotazione:** Il sistema dovrà permettere agli studenti di visualizzare la propria prenotazione.
- **RF8-Rimuovi prenotazione:** Il sistema dovrà permettere agli studenti di cancellare la propria prenotazione.

2.1.4 Gestione Profilo

- **RF9-CRUD Account:** Il sistema dovrà gestire le attività CRUD sui profili utenti.

2.1.5 Gestione Utente

- **RF10-Visualizza utente:** Il sistema dovrà permettere di visualizzare i dettagli relativi a un utente.
- **RF11-Visualizza lista utenti:** Il sistema dovrà permettere al personale della biblioteca di visualizzare gli utenti registrati.

2.1.6 Gestione Sistema

- **RF12-Backup :** Il sistema dovrà effettuare un backup periodico dei dati.
- **RF13-Scadenza prenotazione:** Il sistema dovrà gestire il termine delle prenotazioni.
- **RF14-Scadenza Prestito:** Il sistema dovrà gestire la scadenza dei prestiti.
- **RF15- Autenticazione:** Il sistema dovrà consentire solamente agli utenti registrati di accedere alla piattaforma.

2.2 Requisiti non funzionali

I requisiti non funzionali rappresentano i vincoli imposti al sistema. Sono stati suddivisi in 2 macro-categorie ognuna rappresentata da un package contenente i corrispondenti requisiti come mostrato in figura 2.3

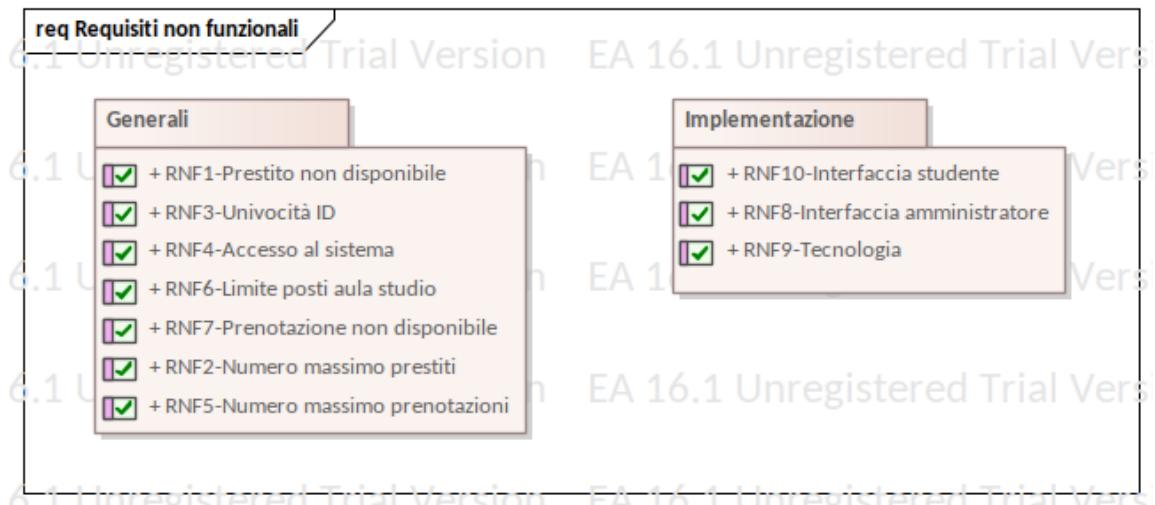


Figura 2.3: Requisiti Funzionali

2.2.1 Generali

- **RNF1-Prestito non disponibile:** Il sistema non dovrà permettere di effettuare il prestito di un documento non disponibile.
- **RNF2-Numero massimo prestiti:** Il sistema non dovrà permettere che uno studente effettui più di due prestiti alla volta.
- **RNF3-Univocità ID:** Il sistema dovrà utilizzare dei codici identificativi univoci: Gli studenti, gli amministratori e le prenotazioni saranno identificati dalla matricola fornita dall’Ateneo. I libri saranno identificati dal proprio ISBN, mentre le riviste saranno identificate dal proprio ISSN.
- **RNF4-Accesso al sistema:** Il sistema dovrà permettere all’utente l’accesso alla piattaforma tramite l’utilizzo di un codice identificativo (matricola) e una password alfanumerica compresa tra gli 8 e i 32 caratteri.

- **RNF5-Numero massimo prenotazioni:** Il sistema non dovrà permettere che uno studente effettui più di una prenotazione alla volta.
- **RNF6-Limite posti aula studio:** Il sistema prenotazioni dovrà tener conto della capienza massima delle aule adibite allo studio.
- **RNF7-Prenotazione non disponibile:** Il sistema non dovrà permettere di effettuare la prenotazione di un turno non disponibile.

2.2.2 Implementazione

- **RNF8-Interfaccia amministratore:** : Il sistema dovrà essere dotato di un'interfaccia grafica comune a tutto il personale della biblioteca.
- **RNF9-Tecnologia:** Il sistema dovrà essere implementato utilizzando Python 3.10
- **RNF10-Interfaccia studente::** Il sistema dovrà essere dotato di un'interfaccia grafica comune a tutti gli studenti.

3. CASI D'USO

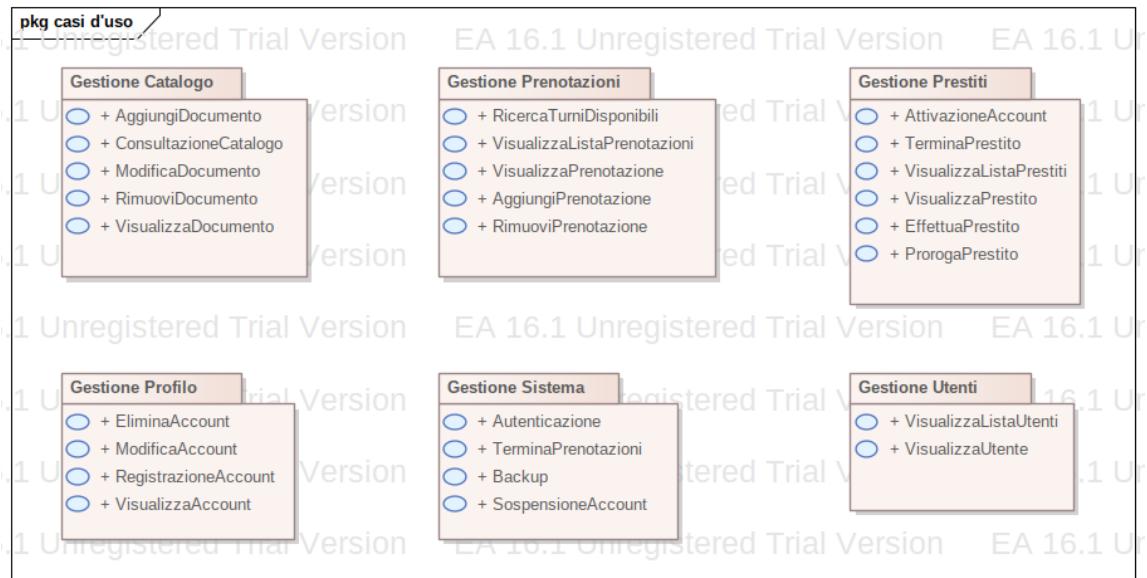


Figura 3.1: Casi d'uso

I casi d'uso del sistema sono stati suddivisi in 6 macro-categorie:

- Gestione Catalogo
- Gestione Prenotazioni
- Gestione Prestiti
- Gestione Profilo
- Gestione Sistema

- Gestione Utenti

Tale suddivisione è identica a quella adottata nel paragrafo 2.1, ognuna concretizzata in un diagramma dei casi d'uso, come rappresentato in figura 3.1

Gli attori che interagiscono con il sistema sono rappresentati in figura 3.2

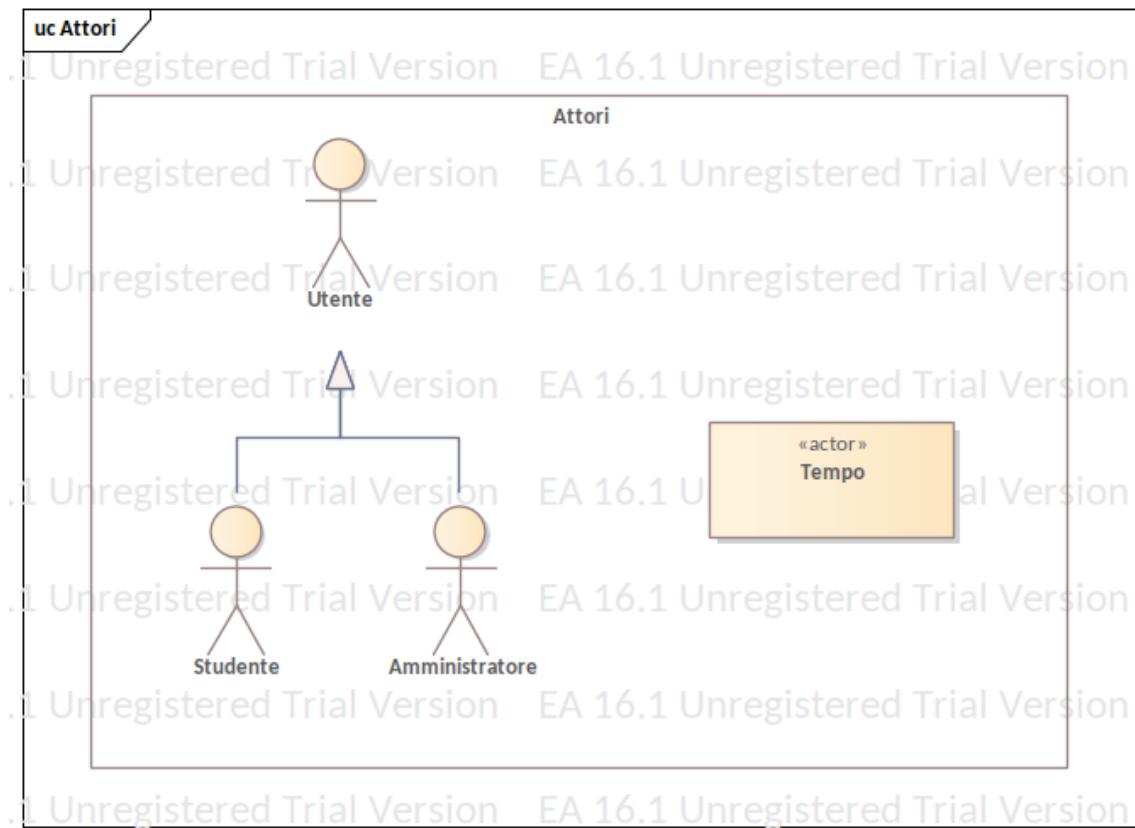


Figura 3.2: Casi d'uso

3.1 Gestione Catalogo

Il diagramma dei casi d'uso mostrato in figura 3.3 rappresenta la gestione dei documenti all'interno del sistema.

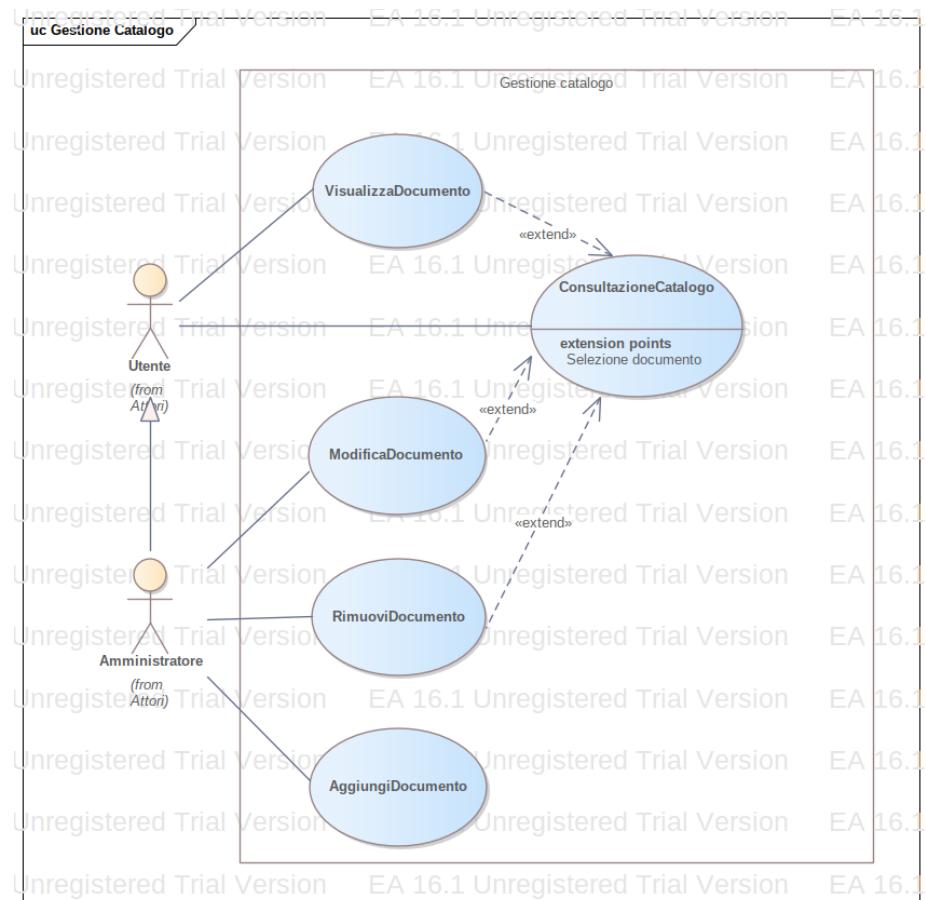


Figura 3.3: Gestione Catalogo

I casi d'uso presenti in questo package sono:

- AggiungiDocumento
- VisualizzaDocumento
- ModificaDocumento
- RimuoviDocumento

- ConsultazioneCatalogo

Gli attori che interagiscono con questi casi d'uso sono l'*Utente* l'*Amministratore*.

3.1.1 AggiungiDocumento

Questo caso d'uso consente di aggiungere un nuovo documento al catalogo della biblioteca.

Attori primari: Amministratore.

Attori secondari: Nessuno.

Precondizioni: L'attore primario è stato autenticato dal sistema.

Postcondizioni: Il documento è stato aggiunto al database della biblioteca.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole aggiungere un nuovo documento all'interno della biblioteca;
2. Il sistema preleva i parametri relativi al documento.
3. *if* i dati inseriti sono validi:
 - 3.1 *if* il codice identificativo del documento da aggiungere non è già associato a un altro documento:
 - 3.1.1 il sistema memorizza il nuovo documento.
 - 3.2 *else:*
 - 3.2.1 il sistema mostra un opportuno messaggio di errore;
4. *else:*
 - 4.1 Il sistema mostra un opportuno messaggio di errore;

Sequenza degli eventi alternativa: Nessuna.

3.1.2 ConsultazioneCatalogo

Questo caso d'uso consente di visualizzare i documenti presenti all'interno della biblioteca che soddisfano i parametri di ricerca.

Attori primari: Utente.

Attori secondari: Nessuno.

Precondizioni: L'attore primario è stato autenticato dal sistema.

Postcondizioni: Il sistema mostra la lista dei documenti che soddisfano i parametri di ricerca.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole consultare il catalogo documenti della biblioteca;
2. L'attore primario specifica i criteri di ricerca;
3. Il sistema ricerca i documenti che soddisfano i parametri specificati;
4. *for each* documento trovato:
 - 4.1 Il sistema aggiunge il documento ai risultati di ricerca;
5. il sistema mostra i risultati di ricerca.

punto di estensione:) selezione documento

Sequenza degli eventi alternativa: Nessuna.

3.1.3 VisualizzaDocumento

Segmento 1: Questo caso d'uso consente di visualizzare le informazioni relative ad un documento.

Attori primari: Utente.

Attori secondari: Nessuno.

Precondizioni del segmento 1: L'attore primario ha selezionato un documento.

Postcondizioni del segmento 1: Il sistema mostra i dettagli relativi al documento.

Sequenza degli eventi del segmento 1:

1. Il sistema preleva la chiave di ricerca del documento;
2. Il sistema recupera il documento dal database della biblioteca;
3. il sistema visualizza i dati relativi al documento.

Sequenza degli eventi alternativa: Nessuna.

3.1.4 ModificaDocumento

Segmento 2: Questo caso d'uso consente di modificare le informazioni relative ad un documento

Attori primari: Amministratore.

Attori secondari: Nessuno.

Precondizioni del segmento 2: L'attore primario ha selezionato un documento.

Postcondizioni del segmento 2: Il sistema salva le modifiche apportate al documento.

Sequenza degli eventi del segmento 2:

1. Il sistema preleva la chiave di ricerca del documento;
2. Il sistema recupera il documento dal database della biblioteca e lo restituisce;
3. L'attore primario modifica i dati relativi al documento.
4. *if* i dati inseriti sono validi:
 - 4.1 il sistema memorizza le modifiche effettuate del documento.
5. *else:*
 - 5.1 Il sistema mostra un opportuno messaggio di errore;

Sequenza degli eventi alternativa: Nessuna.

3.1.5 RimuoviDocumento

Segmento 3: Questo caso d'uso consente di rimuovere un documento all'interno del sistema.

Attori primari: Amministratore.

Attori secondari: Nessuno.

Precondizioni del segmento 3: L'attore primario ha selezionato un documento.

Postcondizioni del segmento 3: Il documento è stato rimosso dal database della biblioteca.

Sequenza degli eventi del segmento 3:

1. Il sistema chiede all'attore primario se desidera rimuovere il documento;
2. *if* l'attore primario conferma l'operazione:
 - 2.1 Il sistema preleva la chiave di ricerca del documento;
 - 2.2 Il sistema provvede alla rimozione del documento dal database della biblioteca.

Sequenza degli eventi alternativa: Nessuna.

3.2 Gestione Prenotazioni

Il diagramma dei casi d'uso mostrato in figura 3.4 rappresenta la gestione delle prenotazioni all'interno del sistema.

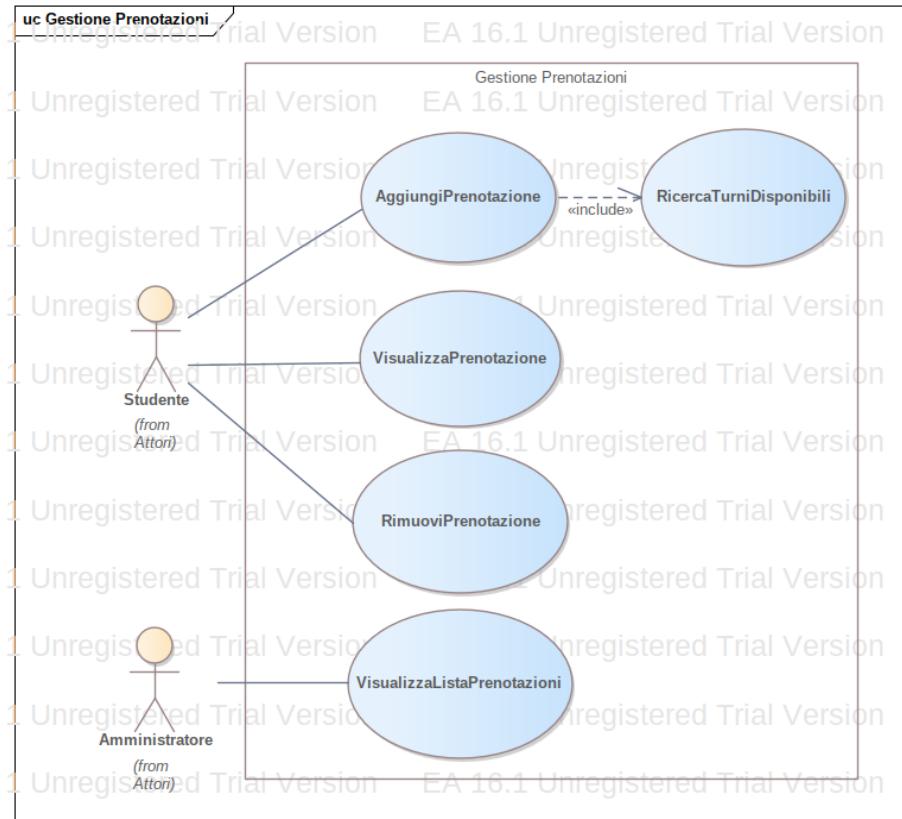


Figura 3.4: Gestione prenotazioni

I casi d'uso presenti in questo package sono:

- AggiungiPrenotazione
- VisualizzaPrenotazione
- RimuoviPrenotazione
- VisualizzaListaPrenotazioni
- RicercaTurniDisponibili

Gli attori che interagiscono con questi casi d'uso sono lo *Studente* e l'*Amministratore*.

3.2.1 AggiungiPrenotazione

Questo caso d'uso consente di aggiungere una nuova prenotazione.

Attori primari: Studente.

Attori secondari: Nessuno.

Precondizioni:

1. L'attore primario è stato autenticato dal sistema.
2. Nel momento della prenotazione almeno un posto aula studio è disponibile.

Postcondizioni: La prenotazione è stata aggiunta.

Sequenza degli eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole aggiungere una nuova prenotazione;
2. *include (RicercaTurniDisponibili)*
3. L'attore primario seleziona e conferma la fascia oraria in cui intende prenotarsi;
4. *if* la fascia oraria contiene massimo 4 turni *and* comprende solo turni consecutivi:
 - 4.1 *if* l'attore primario possiede già una prenotazione:
 - 4.1.1 Il sistema mostra un messaggio in cui chiede all'attore primario se desidera sostituire la prenotazione;
 - 4.2 *else:*
 - 4.2.1 Il sistema aggiunge la prenotazione e mostra un messaggio di conferma.
5. *else:*
 - 5.1 Il sistema visualizza un opportuno messaggio di errore;

Sequenza degli eventi alternativa: Nessuna.

3.2.2 VisualizzaPrenotazione

Questo caso d'uso consente di visualizzare le informazioni relative ad una prenotazione.

Attori primari: Studente.

Attori secondari: Nessuno.

Postcondizioni: Il sistema mostra i dati relativi alla prenotazione.

Precondizioni:

1. L'attore primario è stato autenticato dal sistema;
2. L'attore primario possiede una prenotazione.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi ad una prenotazione;
2. Il sistema preleva la chiave di ricerca della prenotazione;
3. Il sistema recupera la prenotazione dal database della biblioteca;
4. Il sistema visualizza i dati relativi alla prenotazione.

Sequenza degli eventi alternativa: Nessuna.

3.2.3 RimuoviPrenotazione

Questo caso d'uso consente di rimuovere una prenotazione.

Attori primari: Studente.

Attori secondari: Nessuno.

Precondizioni:

1. L'attore primario è stato autenticato dal sistema.
2. L'attore primario possiede una prenotazione.

Postcondizioni: La prenotazione è stata rimossa.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando lo studente vuole rimuovere una prenotazione;
2. Il sistema mostra un messaggio in cui chiede di confermare la rimozione della prenotazione;
3. *if* l'attore primario conferma l'operazione:
 - 3.1 Il sistema preleva la chiave di ricerca della prenotazione
 - 3.2 Il sistema provvede alla rimozione della prenotazione dal database della biblioteca.

Sequenza degli eventi alternativa: Nessuna.

3.2.4 VisualizzaListaPrenotazioni

Questo caso d'uso consente di visualizzare le prenotazioni presenti all'interno della biblioteca che soddisfano i parametri di ricerca.

Attori primari: Amministratore.

Attori secondari: Nessuno.

Precondizioni: L'attore primario è stato autenticato dal sistema.

Postcondizioni: Il sistema visualizza la lista delle prenotazioni che soddisfano i parametri di ricerca.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare le prenotazioni presenti nel sistema della biblioteca;
2. L'attore primario specifica i criteri di ricerca;
3. il sistema ricerca le prenotazioni che soddisfano i parametri specificati;
4. *for each* prenotazione trovata:
 - 4.1 Il sistema aggiunge la prenotazione ai risultati di ricerca;
5. Il sistema mostra i risultati di ricerca.

Sequenza degli eventi alternativa: Nessuna.

3.2.5 RicercaTurniDisponibili

Questo caso d'uso restituisce tutti le fasce orarie in cui è possibile prenotarsi in base alla sede e data specificata.

Attori primari: Studente

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni: Il sistema restituisce i turni disponibili che soddisfano i parametri di ricerca.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando uno studente vuole effettuare una prenotazione;
2. L'attore primario specifica i parametri di ricerca;
3. il sistema ricerca i turni disponibili che soddisfano i criteri specificati;
4. *for each* turno trovato:
 - 4.1 Il sistema aggiunge il turno ai risultati di ricerca;
5. Il sistema restituisce e mostra i risultati di ricerca;

Sequenza degli eventi alternativa: Nessuna.

3.3 Gestione Sistema

Il diagramma dei casi d'uso mostrato in figura 3.5 rappresenta la gestione generale del sistema bibliotecario.

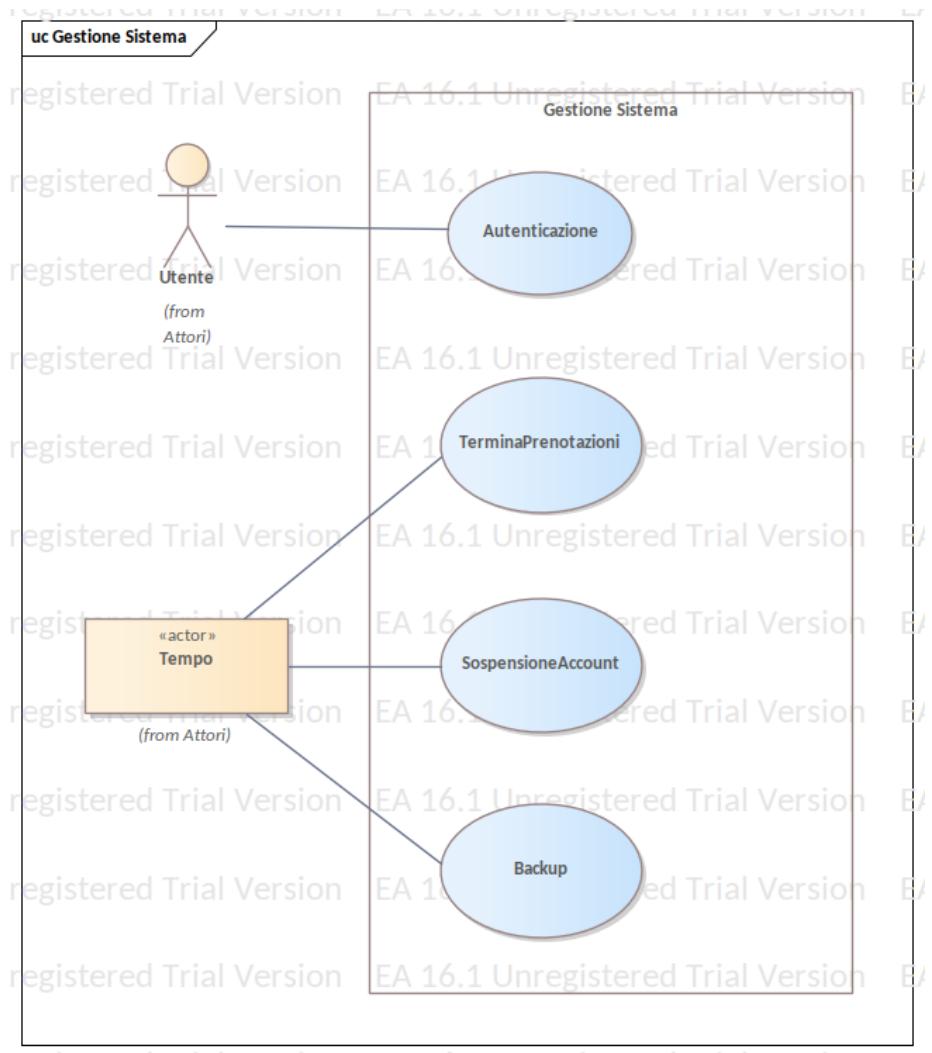


Figura 3.5: Gestione Sistema

I casi d'uso presenti in questo package sono:

- Autenticazione
- Backup

- SospensioneAccount
- TerminaPrenotazioni

Gli attori che interagiscono con questi casi d'uso sono l'*Utente* e il *Tempo*.

3.3.1 Autenticazione

Questo caso d'uso consente l'accesso al sistema da parte dell'Utente.

Attori primari: Utente.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni: L'attore primario è stato autenticato dal sistema.

Sequenza degli eventi principale:

1. Il caso d'uso inizia quando l'utente vuole accedere al sistema;
2. Il sistema preleva i dati inseriti dall'utente;
3. *if* le credenziali inserite sono valide:
 - 3.1 *if* le credenziali inserite sono associate ad un account *and* l'account non è sospeso *and* la chiave d'accesso è corretta:
 - 3.1.1 il sistema permette all'utente di accedere al sistema.
 - 3.2 *else:*
 - 3.2.1 il sistema visualizza un opportuno messaggio di errore;
4. *else:*
 - 4.1 il sistema visualizza un opportuno messaggio di errore;

Sequenza degli eventi alternativa: Nessuna.

3.3.2 Backup

Questo caso d'uso gestisce il backup dei dati relativi al sistema informativo di riferimento.

Attori primari: Tempo.

Attori secondari: Nessuno.

Precondizioni: sono passate 24h dall'ultimo backup.

Postcondizioni: i dati sono stati copiati.

Sequenza degli eventi principale:

1. il caso d'uso inizia quando l'attore primario richiede il backup dei dati;
2. il sistema preleva tutti i dati presenti nel sistema;
3. il sistema copia sul disco i dati prelevati;

Sequenza degli eventi alternativa: Nessuna.

3.3.3 SospensioneAccount

Questo caso d'uso gestisce la sospensione dell'account studente quando quest'ultimo non conclude un prestito entro i termini stabiliti.

Attori primari: Tempo.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni: Lo studente è stato sospeso dalla piattaforma.

Sequenza degli eventi principale:

1. il caso d'uso inizia ogni giorno alle ore 00.00;
2. il sistema preleva tutti i prestiti presenti nel database della biblioteca;
3. *for each* prestito:
 - 3.1 *if* la data di scadenza del prestito è passata:
 - 3.1.1 Il sistema sospende l'account studente a cui è associato il prestito.

Sequenza degli eventi alternativa: Nessuna.

3.3.4 TerminaPrenotazioni

Questo caso d'uso consente di gestire la scadenza di una prenotazione.

Attori primari: Tempo.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni: Le prenotazioni scadute sono state rimosse dal database della biblioteca.

Sequenza degli eventi principale:

1. il caso d'uso inizia ogni giorno alle ore 19.30;
2. il sistema preleva tutte le prenotazioni presenti nel database della biblioteca;
3. *for each* prenotazione:
 - 3.1 *if* la data della prenotazione corrisponde al giorno corrente *and* l'ora di andata è passata:
 - 3.1.1 Il sistema rimuove la prenotazione dal database della biblioteca.

Sequenza degli eventi alternativa: Nessuna.

3.4 Gestione Prestiti

Il diagramma dei casi d'uso mostrato in figura 3.6 rappresenta la gestione dei prestiti all'interno del sistema bibliotecario.

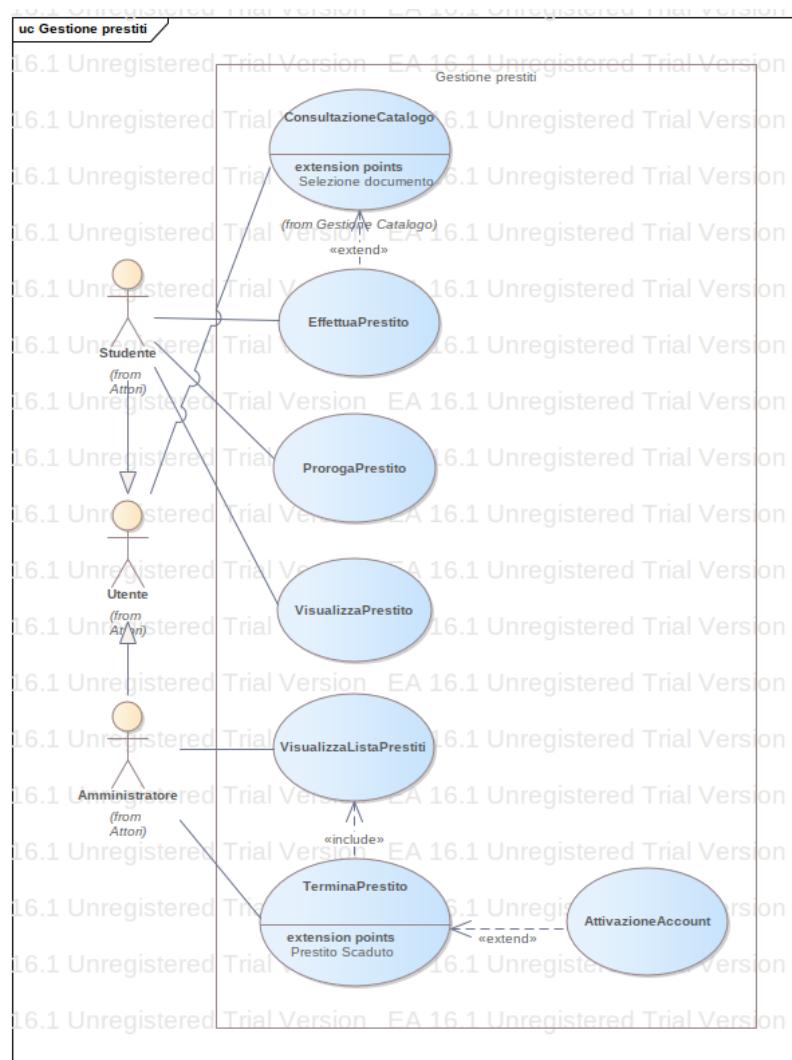


Figura 3.6: Gestione Prestiti

I casi d'uso presenti in questo package sono:

- EffettuaPrestito

- VisualizzaPrestito
- ProrogaPrestito
- TerminaPrestito
- AttivazioneAccount
- VisualizzaListaPrestiti

Gli attori che interagiscono con questi casi d'uso sono l'*Amministratore* e lo *Studente*.

3.4.1 EffettuaPrestito

Segmento 1: Questo caso d'uso consente di effettuare il prestito di un documento presente all'interno della biblioteca.

Attori primari: Studente

Attori secondari: Nessuno

Precondizioni del segmento 1: L'attore primario ha selezionato un documento.

Postcondizioni del segmento 1: Il prestito è stato effettuato.

Sequenza degli eventi principale:

1. Il sistema preleva il codice identificativo associato all'attore primario;
2. *if* il numero di prestiti associati all'attore primario è minore di 2 *and* il documento è disponibile:
 - 2.1 il sistema preleva il codice identificativo associato al documento selezionato;
 - 2.2 *if* il documento selezionato non è stato già preso in prestito:
 - 2.2.1 il sistema effettua il prestito del documento selezionato.
 - 2.3 *else:*
 - 2.3.1 il sistema visualizza un messaggio di errore;
3. *else:*
 - 3.1 il sistema mostra un opportuno messaggio di errore;

Sequenza degli eventi alternativa: Nessuna.

3.4.2 VisualizzaPrestito

Questo caso d'uso consente di visualizzare le informazioni relative ad un prestito.

Attori primari: Studente

Attori secondari: Nessuno

Precondizioni:

1. L'attore primario è stato autenticato dal sistema.
2. L'attore primario possiede un prestito.

Postcondizioni: Il sistema visualizza il prestito.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi ad un prestito;
2. Il sistema preleva la chiave di ricerca del prestito;
3. Il sistema recupera il prestito dal database della biblioteca;
4. Il sistema visualizza i dati relativi al prestito.

Sequenza degli eventi alternativa: Nessuna.

3.4.3 ProrogaPrestito

Questo caso d'uso consente di prolungare la scadenza di un prestito;

Attori primari: Studente.

Attori secondari: Nessuno.

Precondizioni:

1. L'attore primario è stato autenticato dal sistema.
2. L'attore primario possiede un prestito che non è stato precedentemente prorogato.

Postcondizioni: La proroga del prestito è stata effettuata.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole prolungare la scadenza di un prestito;
2. Il sistema preleva la chiave di ricerca del prestito;
3. Il sistema recupera il prestito dal database della biblioteca e ne prolunga di una settimana la scadenza.

Sequenza degli eventi alternativa: Nessuna.

3.4.4 TerminaPrestito

Questo caso d'uso consiste nella riconsegna del documento da parte dello studente al personale della biblioteca e nella conseguente conclusione del prestito.

Attori primari: Amministratore.

Attori secondari: Nessuno.

Precondizioni: L'attore primario è stato autenticato dal sistema.

Postcondizioni: Il prestito è terminato.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando uno studente riconsegna un documento precedentemente preso in prestito al personale della biblioteca;
2. Lo studente fornisce all'attore primario il codice identificativo del prestito;
3. *include (VisualizzaListaPrestiti)*
4. L'attore primario seleziona e conferma il prestito da concludere;
punto di estensione:) prestito scaduto
5. il sistema provvede a incrementare la disponibilità del documento precedentemente preso in prestito;
6. il sistema rimuove il prestito associato allo studente dal database della biblioteca;

Sequenza degli eventi alternativa: Nessuna

3.4.5 AttivazioneAccount

segmento 1: Il sistema provvede alla riattivazione dell'account associato allo studente precedentemente sospeso a causa della mancata restituzione del prestito entro i termini stabiliti.

Attori primari: Amministratore

Attori secondari: Nessuno

Precondizioni del segmento 1: Il prestito è scaduto.

Postcondizioni del segmento 1: L'account associato allo studente è stato riattivato.

Sequenza degli eventi del segmento 1:

1. il sistema preleva la chiave di ricerca dello studente;
2. Il sistema recupera il profilo studente dal database della biblioteca e lo riabilita a poter utilizzare la piattaforma.

Sequenza degli eventi alternativa: Nessuna

3.4.6 VisualizzaListaPrestiti

Questo caso d'uso consente di visualizzare i prestiti presenti all'interno della biblioteca che soddisfano i parametri specificati.

Attori primari: Amministratore.

Attori secondari: Nessuno.

Precondizioni: L'attore primario è stato autenticato dal sistema.

Postcondizioni: Il sistema mostra tutti i prestiti presenti all'interno del sistema che soddisfano i parametri richiesti.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i prestiti presenti all'interno del sistema;
2. L'attore primario specifica i criteri di ricerca;
3. il sistema ricerca i prestiti che soddisfano i parametri specificati;
4. *for each* prestito trovato:
 - 4.1 Il sistema aggiunge il prestito ai risultati di ricerca;
5. il sistema mostra i risultati di ricerca.

Sequenza degli eventi alternativa: Nessuna.

3.5 Gestione Utente

Il diagramma dei casi d'uso mostrato in figura 3.7 rappresenta la gestione degli utenti del sistema bibliotecario.

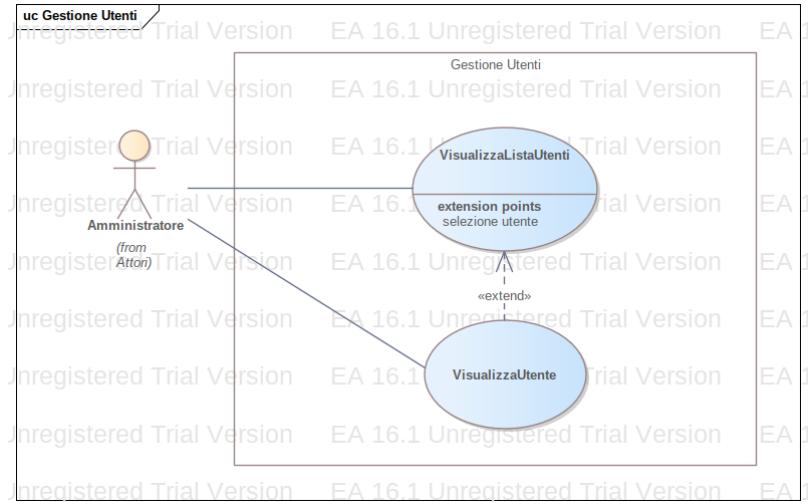


Figura 3.7: Gestione Utenti

I casi d'uso presenti in questo package sono:

- VisualizzaUtente
- VisualizzaListaUtenti

L'attore che interagisce con questo casi d'uso è l'*Amministratore*.

3.5.1 VisualizzaListaUtenti

Questo caso d'uso consente di visualizzare gli utenti presenti nel database della biblioteca che soddisfano i parametri di ricerca.

Attori primari: Amministratore.

Attori secondari: Nessuno.

Precondizioni: L'attore primario è stato autenticato dal sistema.

Postcondizioni: Il sistema visualizza la lista di utenti che soddisfano i criteri di ricerca.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare gli utenti presenti nel sistema della biblioteca;
2. L'attore primario specifica i parametri di ricerca;
3. Il sistema ricerca nel database della biblioteca gli utenti che soddisfano i criteri di ricerca specificati;
4. *for each* utente trovato:
 - 4.1 Il sistema aggiunge l'utente ai risultati di ricerca;
5. Il sistema mostra i risultati di ricerca.

punto di estensione:) Visualizza utente

Sequenza degli eventi alternativa: Nessuna .

3.5.2 VisualizzaUtente

Segmento 1: Questo caso d'uso consente di visualizzare le informazioni relative ad un utente.

Attori primari: Amministratore.

Attori secondari: Nessuno.

Precondizioni del segmento 1: Un utente è stato selezionato dall'attore primario;

Postcondizioni del segmento 1: Il sistema mostra i dati relativi all'utente.

Sequenza degli eventi del segmento 1:

1. il sistema preleva dal database della biblioteca le informazioni associate all'utente selezionato.
2. il sistema mostra i dettagli relativi all'utente.

Sequenza degli eventi alternativa: Nessuna .

3.6 Gestione Profilo

Il diagramma dei casi d'uso mostrato in figura 3.8 rappresenta la gestione degli utenti del sistema bibliotecario.

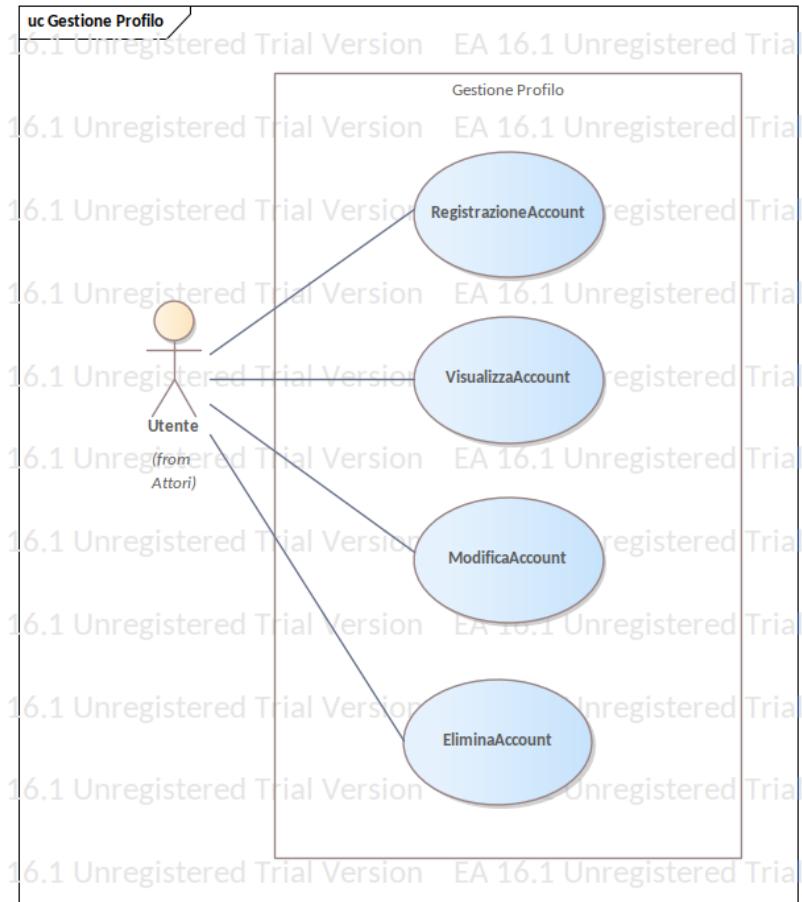


Figura 3.8: Gestione Profilo

I casi d'uso presenti in questo package sono:

- RegistrazioneAccount
- VisualizzaAccount
- ModificaAccount
- EliminaAccount

L'attore che interagisce con questo casi d'uso è l'*Utente*.

3.6.1 VisualizzaAccount

Questo caso d'uso consente di visualizzare le informazioni relative al profilo utente della biblioteca.

Attori primari: Utente.

Attori secondari: Nessuno.

Precondizioni: L'attore primario è stato autenticato dal sistema.

Postcondizioni: Il sistema mostra i dettagli relativi al proprio account.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dettagli relativi al proprio account;
2. Il sistema recupera dal database della biblioteca le informazioni relative all'utente;
3. il sistema mostra i dettagli del profilo utente.

Sequenza degli eventi alternativa: Nessuna.

3.6.2 ModificaAccount

Questo caso d'uso consente di modificare il profilo utente della biblioteca.

Attori primari: Utente.

Attori secondari: Nessuno.

Precondizioni: L'attore primario è stato autenticato dal sistema.

Postcondizioni: Il sistema salve le modifiche apportate all'account.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole modificare i dettagli relativi al proprio account;
2. Il sistema recupera dal database della biblioteca le informazioni relative all'utente;
3. L'attore primario modifica e conferma i dati relativi al proprio account.
4. *if* i dati inseriti sono validi:
 - 4.1 il sistema memorizza le modifiche effettuate del proprio account.
5. *else:*
 - 5.1 Il sistema mostra un opportuno messaggio di errore;

Sequenza degli eventi alternativa: Nessuna.

3.6.3 EliminaAccount

Questo caso d'uso consente di eliminare il profilo utente dal sistema della biblioteca.

Attori primari: Utente.

Attori secondari: Nessuno.

Precondizioni: L'attore primario è stato autenticato dal sistema.

Postcondizioni: Il profilo utente è stato rimosso dal database della biblioteca.

Sequenza degli eventi principale:

1. Il caso d'uso ha inizio quando l'attore primario vuole eliminare il proprio profilo utente;
2. Il sistema chiede all'attore primario di confermare l'eliminazione del proprio account;
3. *if* l'attore primario conferma l'operazione:
 - 3.1 *if* l'attore primario è uno studente *and* al suo profilo è associato almeno un prestito:
 - 3.1.1 Il sistema mostra un opportuno messaggio di errore.
 - 3.2 *else:*
 - 3.2.1 Il sistema provvede alla rimozione dell'utente dal database della biblioteca.

Sequenza degli eventi alternativa: Nessuna.

3.6.4 RegistrazioneAccount

Questo caso d'uso consente all'Utente di registrarsi nella piattaforma.

Attori primari: Utente

Attori secondari: Nessuno

Precondizioni: Nessuna

Postcondizioni:

1. L'account utente è stato creato.
2. L'attore primario è stato autenticato dal sistema.

Sequenza degli eventi principale:

1. Il caso d'uso inizia quando l'utente vuole creare un nuovo account;
2. Il sistema preleva i dati inseriti dall'utente;
3. *if* i dati inseriti sono validi:
 - 3.1 *if* l'utente non dispone già di un account:
 - 3.1.1 Il sistema procede alla creazione del nuovo account;
 - 3.1.2 il sistema permette all'utente di accedere al sistema;
 - 3.2 *else:*
 - 3.2.1 il sistema visualizza un opportuno messaggio di errore;
4. *else:*
 - 4.1 Il sistema mostra un opportuno messaggio di errore;

Sequenza degli eventi alternativa: Nessuna.

4. MATRICE DI MAPPING

In figura 4.1 viene illustrata la *Matrice di Mapping* la quale garantisce che ogni requisito funzionali proposto dal software venga soddisfatto da almeno un caso d'uso.

Source	Target													
	:RF1-Consultazione Catalogo	:RF10-Visualizza utente	:RF11-Visualizza lista utenti	:RF12-Backup	:RF13-Scadenza prenotazione	:RF14-Effettua Prestito	:RF15-Autenticazione	:RF2-CRUD Documento	:RF4-Visualizza lista prestiti	:RF5-Visualizza lista prenotazioni	:RF6-AggIungi prenotazione	:RF7-Visualizza prenotazione	:RF8-Rimuovi prenotazione	:RF9-CRUD Account
:AggiungiDocumento														
:AggiungiPrenotazione														
:AttivazioneAccount														
:Autenticazione														
:Backup														
:ConsultazioneCatalogo	↑													
:EffettuaPrestito														
:EliminaAccount												↑		
:ModificaAccount												↑		
:ModificaDocumento														
:ProrogaPrestito														
:RegistrazioneAccount												↑		
:RimuoviDocumento														
:RimuoviPrenotazione												↑		
:SospensioneAccount														
:TerminaPrenotazioni														
:TerminaPrestito														
:VisualizzaAccount												↑		
:VisualizzaDocumento														
:VisualizzaListaPrenotazioni														
:VisualizzaListaPrestiti														
:VisualizzaListaUtenti		↑												
:VisualizzaPrenotazione												↑		
:VisualizzaPrestito														
:VisualizzaUtente		↑												

Figura 4.1: Matrice di mapping

5. CLASSI DI ANALISI

In figura 5.1 viene illustrato il *diagramma delle classi di analisi* che modella i concetti chiave del progetto. Si è sviluppato così un modello iniziale del software che è stato successivamente approfondito in fase di progettazione.

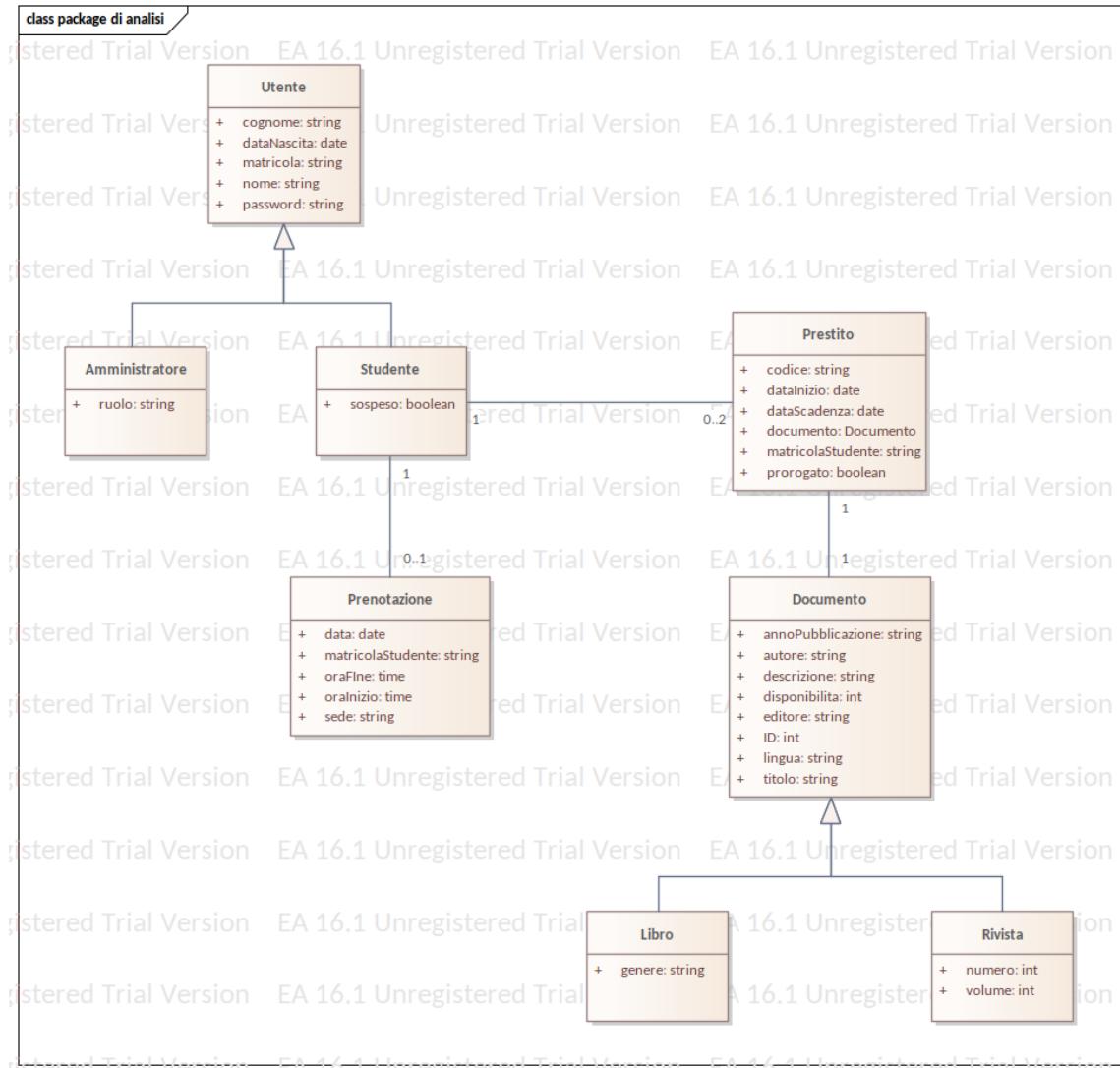


Figura 5.1: Classi di analisi

6. CLASSI DI PROGETTAZIONE

Partendo dal modello rappresentato dalle classi di analisi, nelle *classi di progettazione* si sono definite in modo completo le funzionalità implementate dal software. Si è utilizzato il pattern architetturale Model-View-Controller come illustrato in figura 6.1.

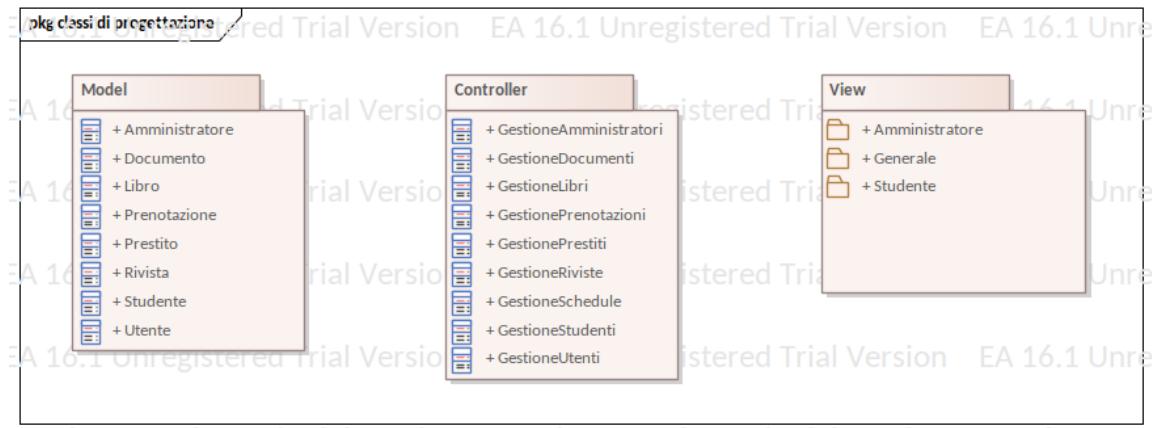


Figura 6.1: Package classi di progettazione

Le classi sono state suddivise nei seguenti package:

- Model
- Controller
- View:
 - Amministratore
 - Generale
 - Studente

6.1 Model

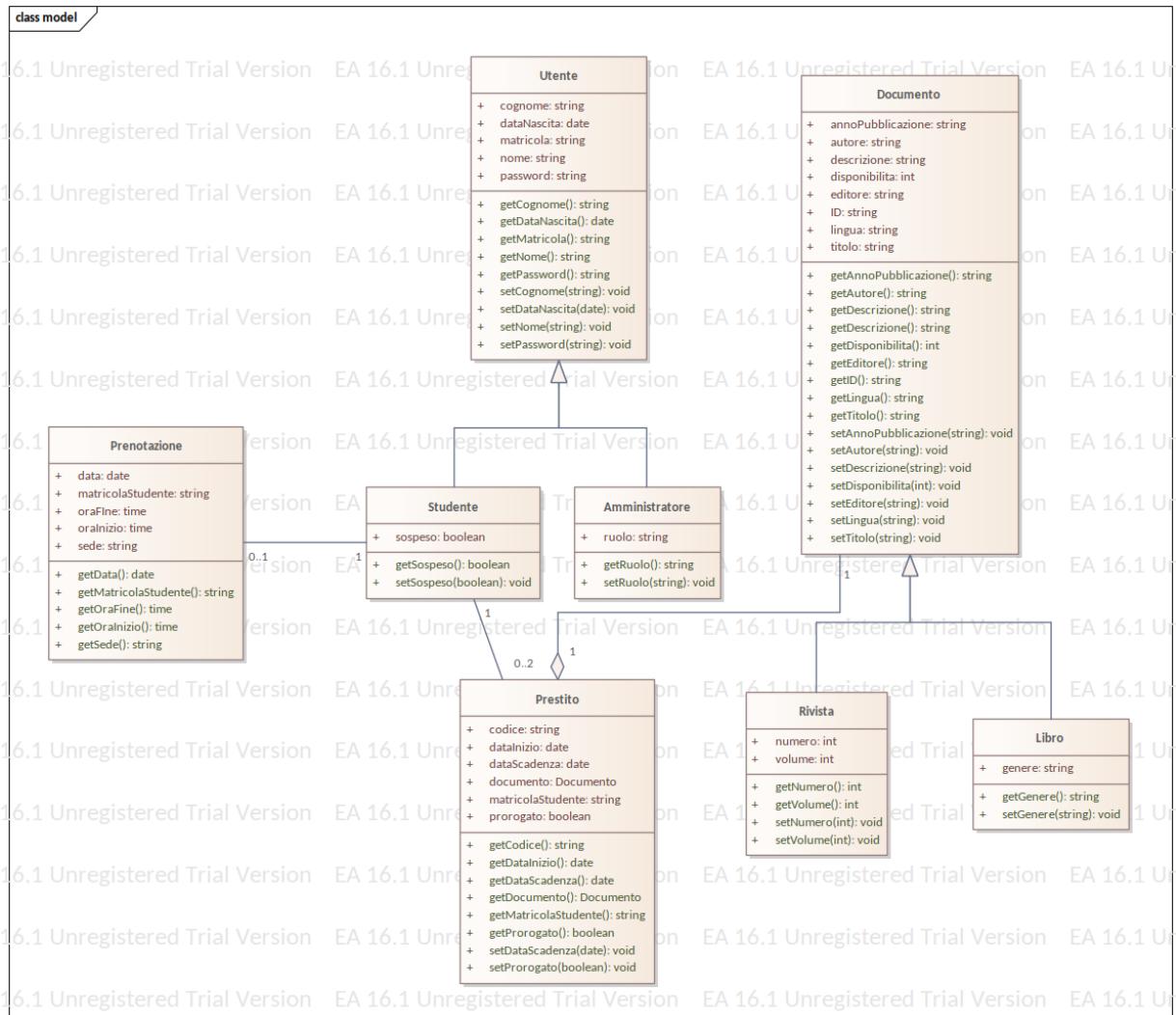


Figura 6.2: Model

6.2 Controller

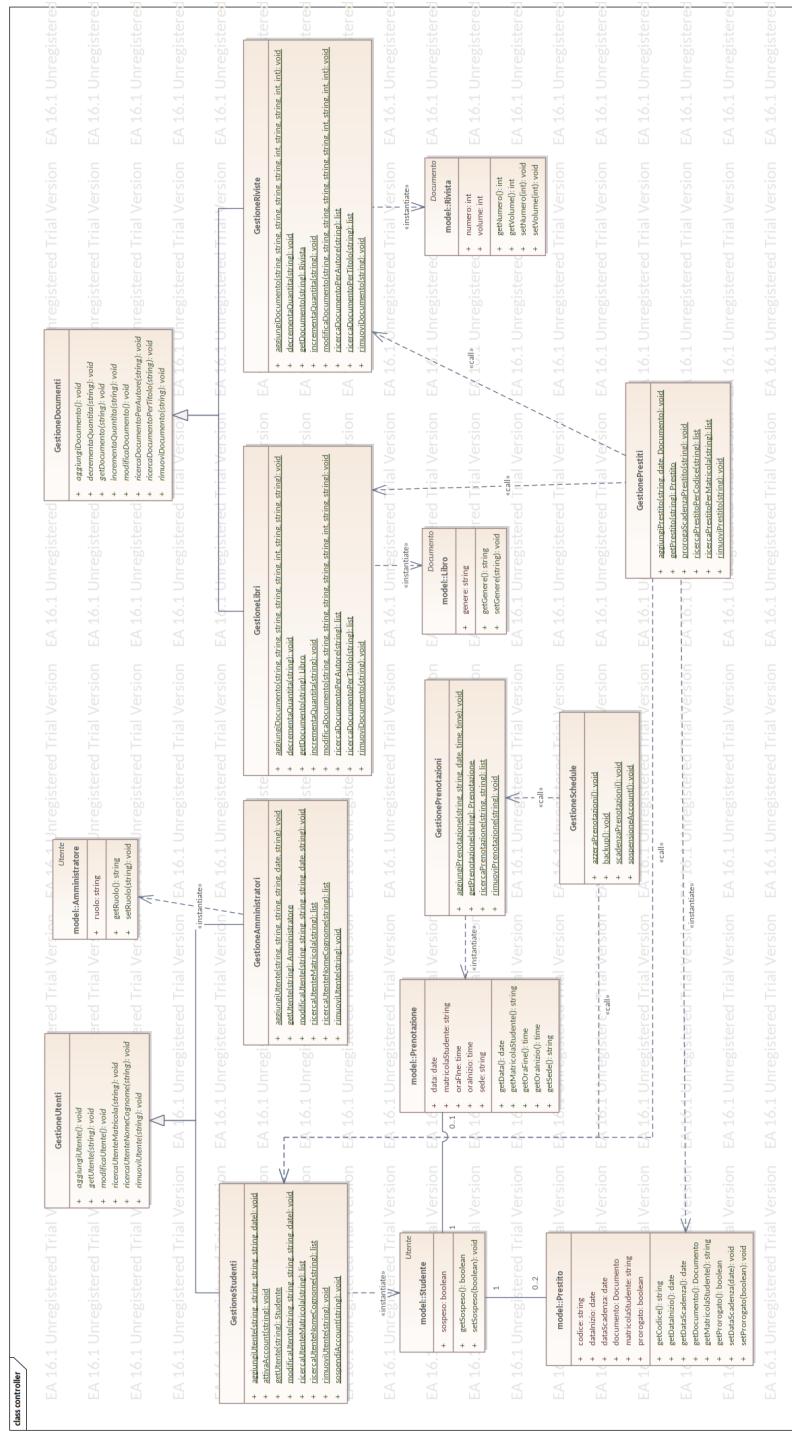


Figura 6.3: Controller

6.3 View

Le interfacce utente sono state realizzate utilizzando QtDesigner, tool offerto da PyQt5. Il package View è suddiviso in ulteriori packages come illustrato in 6.4:

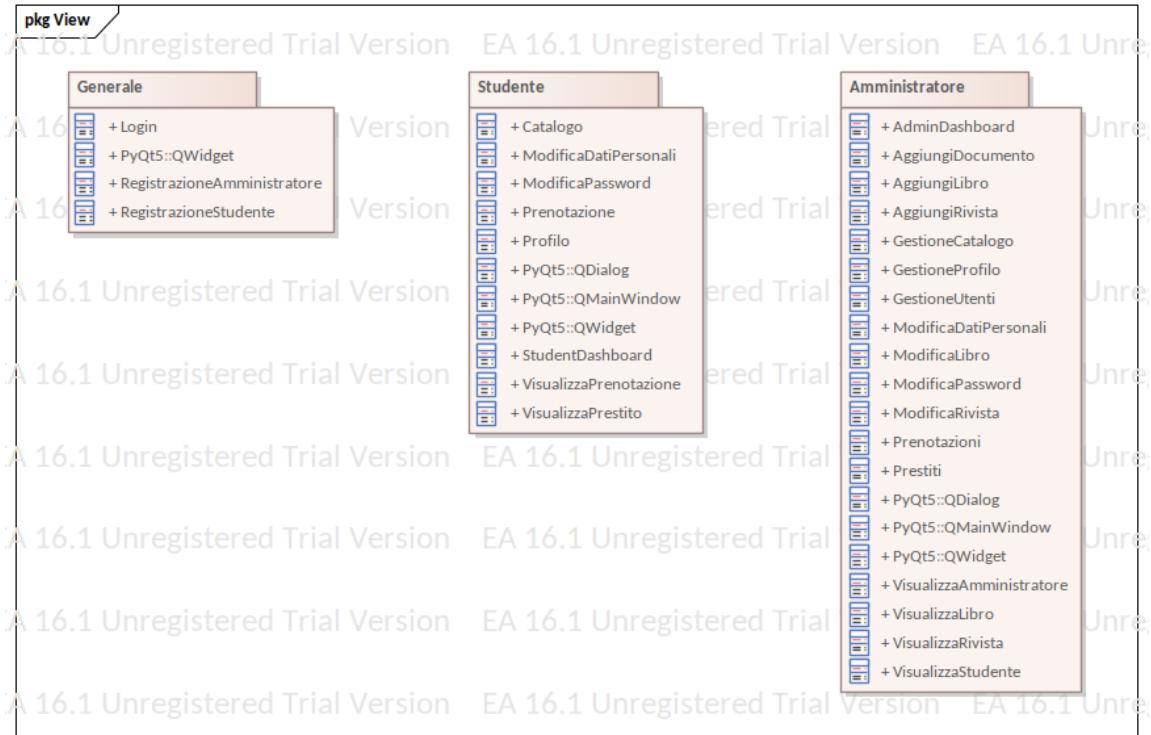


Figura 6.4: package View

Le classi sono state suddivise nei seguenti package:

- Amministratore
- Generale
- Studente

6.3.1 Generale

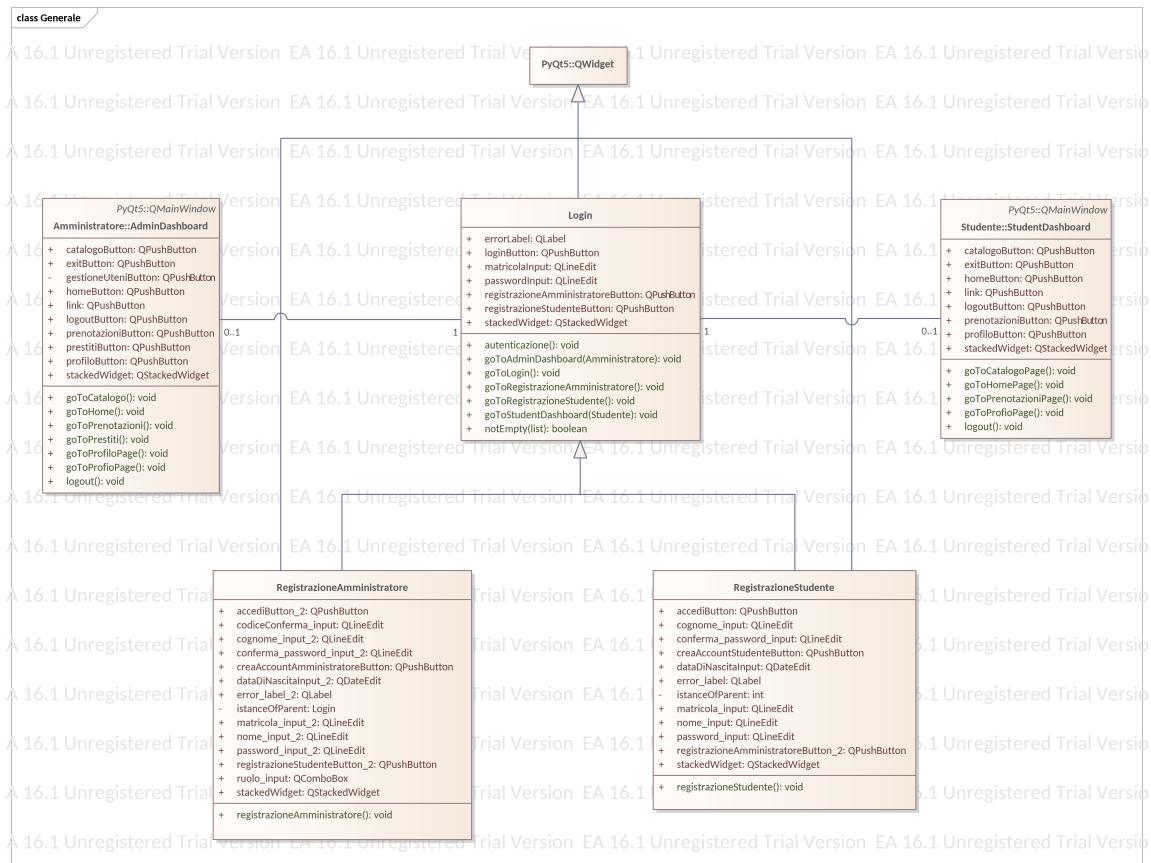


Figura 6.5: Generale

Come mostrato in 6.5 il package *Generale* include le seguenti classi ognuna delle quali importa un file .ui (User Interface):

- Login
- RegistrazioneStudente
- RegistrazioneAmministratore

Login

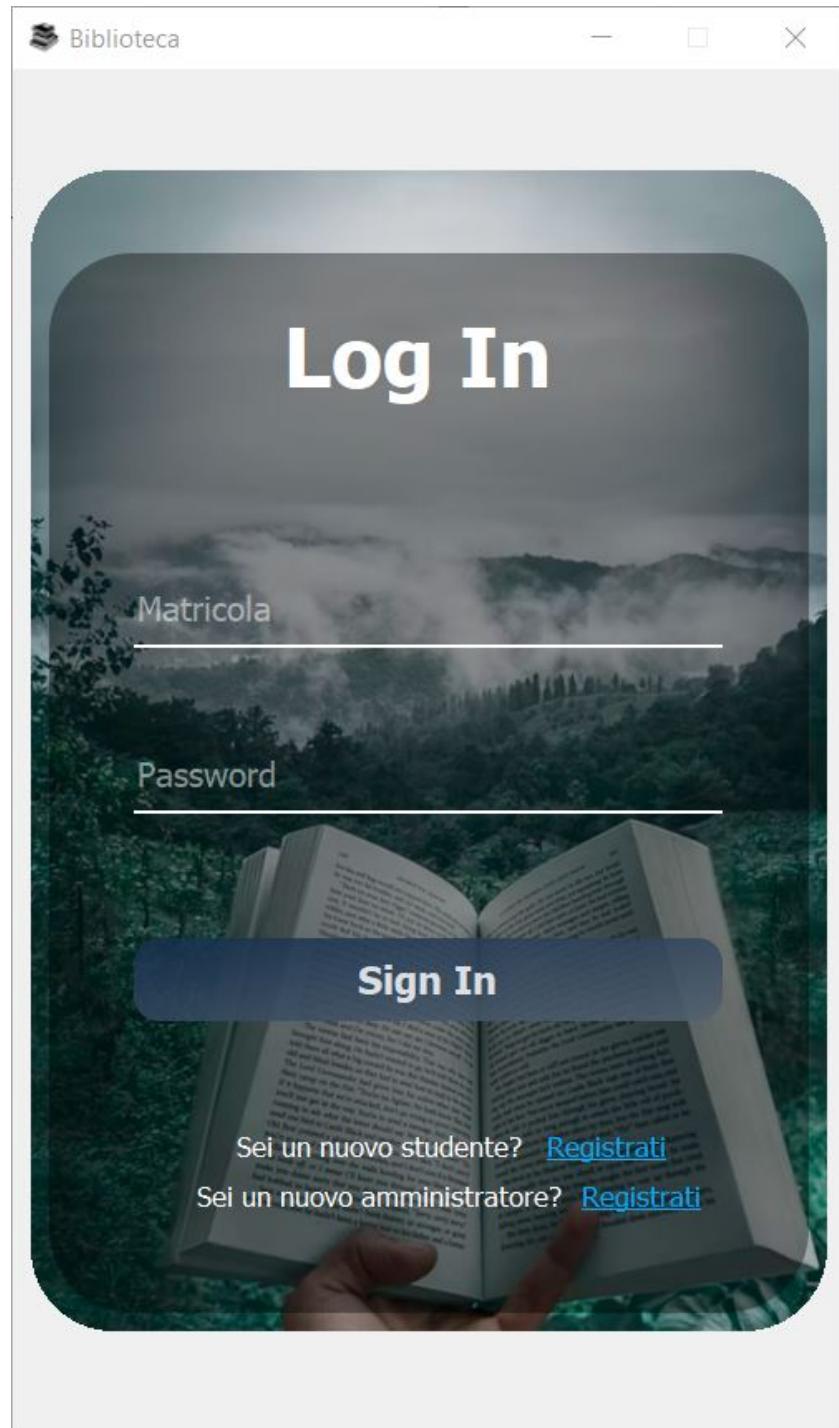


Figura 6.6: Login

RegistrazioneStudente

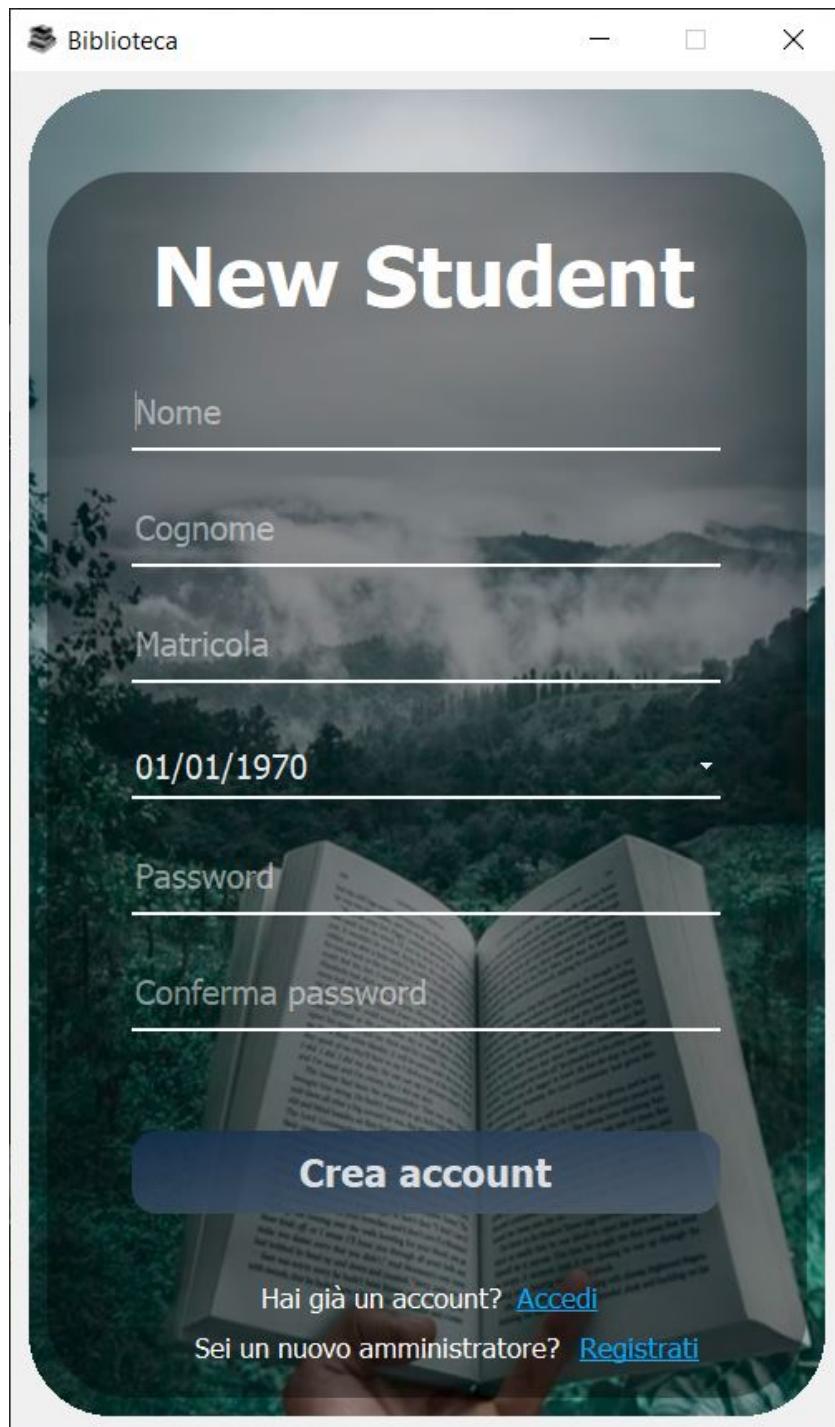


Figura 6.7: RegistrazioneStudente

RegistrazioneAmministratore

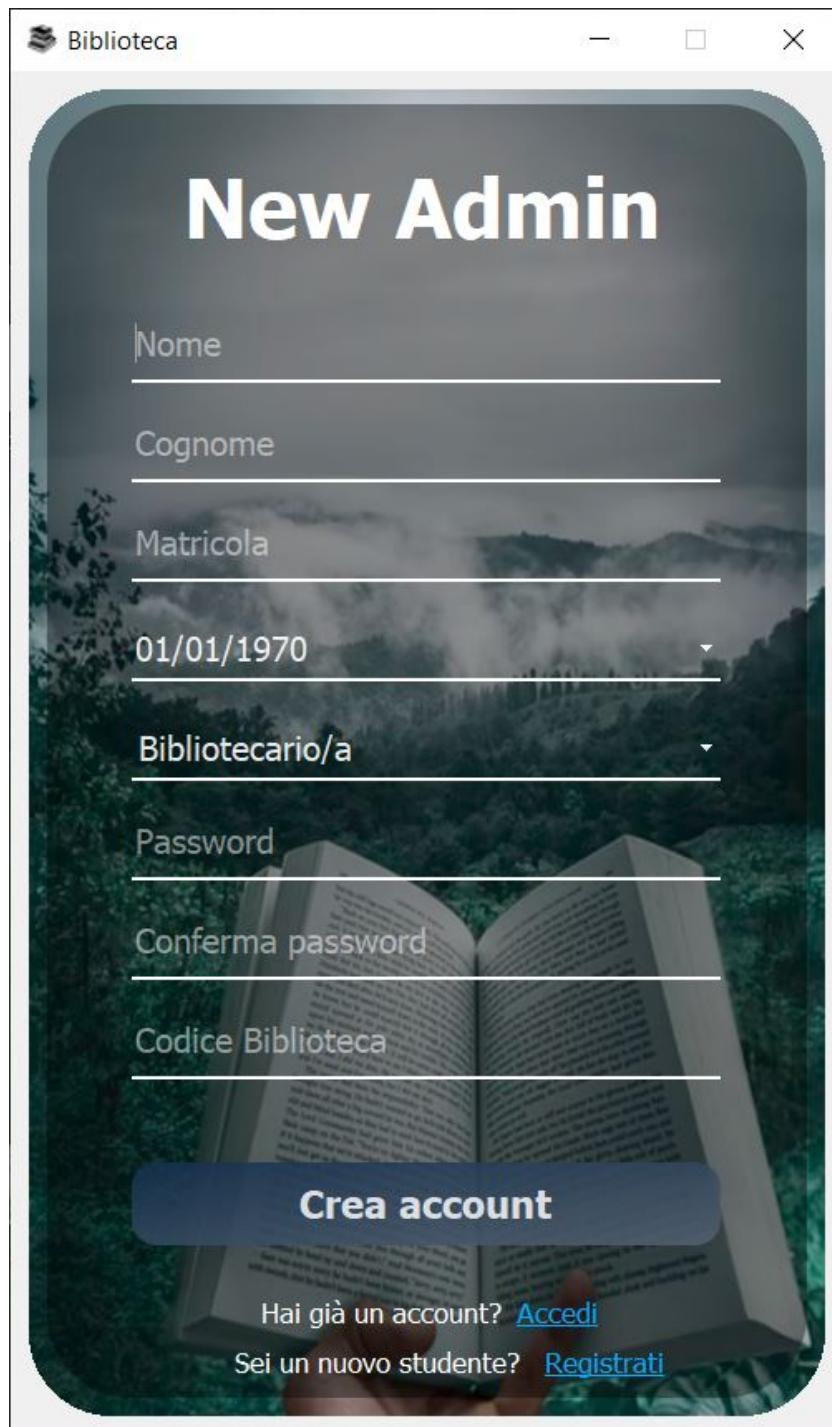


Figura 6.8: RegistrazioneAmministratore

6.3.2 Studente

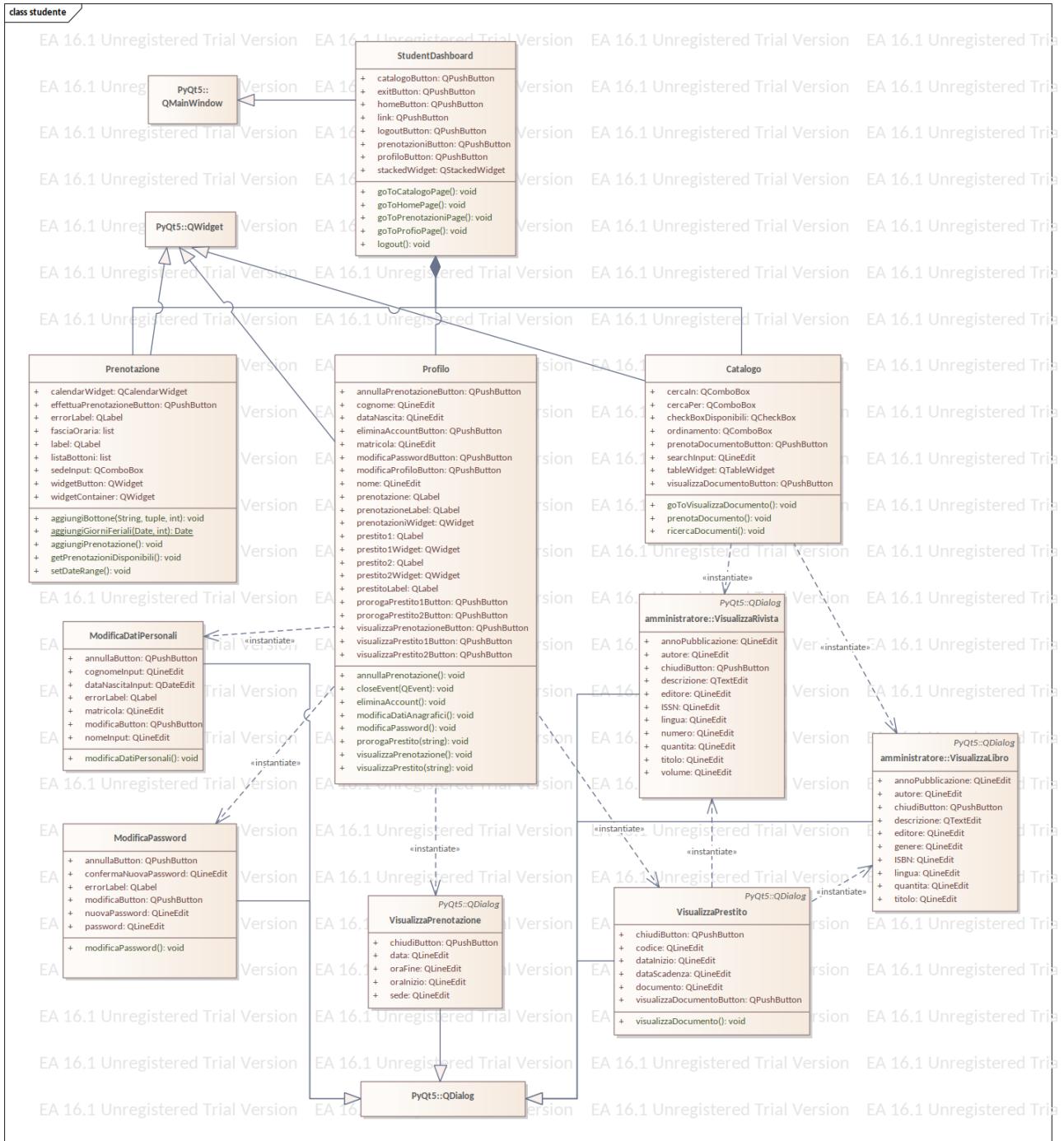


Figura 6.9: Studente

Come mostrato in 6.9 il package *Studente* include le seguenti classi ognuna delle quali importa un file .ui (User Interface):

- StudentDashboard
- Catalogo
- Prenotazione
- Profilo
- ModificaDatiPersonalni
- ModificaPassword
- VisualizzaPrenotazione
- VisualizzaPrestito

StudentDashboard

The screenshot shows the StudentDashboard application window. On the left, there is a vertical sidebar with a yellow background containing several menu items: HOME, CATALOGO, PRENOTAZIONI, PROFILO, LOGOUT, and EXIT. At the top of the sidebar is the logo of the Università Politecnica delle Marche (UNIVPM) Biblioteca, which features a stack of books and the text "Biblioteca UNIVPM". The main content area has a white background. At the top right, it says "Benvenuto!". Below this, there is a paragraph of text about the library system's purpose and features. To the right of the text is a code snippet in Python. Further down, there is a section titled "Chi siamo" with information about the software's creators and their university. On the far right, the university's logo is displayed.

Biblioteca

Biblioteca
UNIVPM

HOME

CATALOGO

PRENOTAZIONI

PROFILO

LOGOUT

EXIT

Benvenuto!

Il Sistema Bibliotecario di Ateneo propone a tutti gli studenti un ricco patrimonio bibliografico. La seguente piattaforma consente agli studenti di:

- Effettuare prestito librario;
- Prenotare un posto aula studio;

Inoltre la piattaforma aiuta il personale nella gestione della biblioteca.

Il software

```
31     self.filat = []
32     self.settings = None
33     self.logined = False
34     self.username = None
35     self.password = None
36
37     self.url = "http://127.0.0.1:5000"
38
39     self.headers = {
40         "Content-Type": "application/json",
41         "Accept": "application/json"
42     }
43
44     def __init__(self, username, password):
45         self.username = username
46         self.password = password
47
48         self.login()
49
50     def login(self):
51         self.logined = True
52
53     def get_settings(self):
54         return self.settings
55
56     def set_settings(self, settings):
57         self.settings = settings
58
59     def request(self, method, url, params=None, data=None):
60         if method == "GET":
61             response = requests.get(url, params=params)
62         else:
63             response = requests.post(url, json=data)
64
65         return response.json()
66
67     def request_login(self, username, password):
68         self.logined = True
69         self.username = username
70         self.password = password
71
72         self.login()
73
74     def request_logout(self):
75         self.logined = False
76
77         self.logout()
78
79     def logout(self):
80         self.logined = False
81
82         self.username = None
83         self.password = None
84
85         self.login()
```

La piattaforma è stata realizzata in python 3.10. La GUI è stata realizzata utilizzando PyQt5. Il codice sorgente e la descrizione dettagliata del progetto sono disponibili su GitHub al seguente link:
[GitHub - Biblioteca-universitaria](#)

Chi siamo

Il software è stato realizzato da FRANCESCO FINIZII e KRISTIAN LIKAJ, studenti di Ingegneria informatica e dell'automazione presso UNIVPM



Figura 6.10: StudentDashboard

Catalogo

CATALOGO

Titolo ▾ Tutto il catalogo ▾

Ordinamento: Titolo A-Z ▾ Solo disponibili

ID	Titolo	Autore	Disponibilità
1	C e C++	Massimo ...	8849407424
2	Follia	Patrick McGrath	1234567891234
3	guida PHP	R. Nikolassy	8820334887
4	spider-man	stan lee	12345678

Prenota

Visualizza

| |

Figura 6.11: Catalogo

Prenotazione

Prenotazione sala studio

Benvenuto studente!

In questa sezione è possibile effettuare una prenotazione per le aule studio presenti all'interno delle biblioteche di ateneo.

E' consentita una sola prenotazione per volta.

Una prenotazione può durare al massimo 2 ore, dunque è possibile selezionare al massimo 4 fasce orarie consecutive.

Sede Biblioteca Economico-Giuridica-Sociologica ▾

ottobre 2023						
lun	mar	mer	gio	ven	sab	dom
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Selezione una fascia oraria:

08:30-09:00 09:00-09:30 09:30-10:00
10:00-10:30 10:30-11:00 11:00-11:30
11:30-12:00 12:00-12:30 12:30-13:00
13:00-13:30 13:30-14:00 14:00-14:30
14:30-15:00 15:00-15:30 15:30-16:00
16:00-16:30 16:30-17:00 17:00-17:30
17:30-18:00 18:00-18:30 18:30-19:00

Effettua prenotazione

Figura 6.12: Prenotazione

Profilo

Informazioni personali

Matricola:

Nome:

Cognome:

Data di nascita:

[Modifica dati personali](#) [Cambia password](#) [Elimina account](#)

Prestiti

Prestito n° 3 [Visualizza](#) [Proroga](#)

Prestito n° 4 [Visualizza](#) [Proroga](#)

Prenotazioni

Prenotazione [Visualizza](#) [Annulla](#)

Figura 6.13: Profilo

ModificaDatiPersonali

 Modifica dati personali ? 

Matricola:	<input type="text" value="123456789"/>
Nome:	<input type="text" value="studente"/>
Cognome:	<input type="text" value="studente"/>
Data di nascita:	<input type="text" value="01-01-2000"/> 

Figura 6.14: ModificaDatiPersonali

ModificaPassword

The screenshot shows a window titled "Cambia password" with a logo of a stylized "S" icon. In the top right corner are buttons for help (?) and close (X). The main area contains three input fields: "Password precedente:" with an empty input field, "Nuova password:" with an empty input field, and "Conferma nuova password:" with an empty input field. At the bottom are two buttons: "Modifica" with a pencil icon and "Annulla" with a red X icon.

Cambia password	?	X
Password precedente:	<input type="text"/>	
Nuova password:	<input type="text"/>	
Conferma nuova password:	<input type="text"/>	
Modifica		Annulla

Figura 6.15: ModificaPassword

VisualizzaPrenotazione

Visualizza prenotazione

Sede	Biblioteca Economico-Giuridica-Sociologica
Data:	17/10/23
Ora arrivo:	08:30
Ora andata	09:30

Figura 6.16: VisualizzaPrenotazione

VisualizzaPrestito

Visualizza Prestito ? X

Prestito n°	3
Effettuato il:	17/10/2023
Valido fino al:	16/11/2023
ID Documento:	8849407424 
<input type="button" value="Chiudi"/>	

Figura 6.17: VisualizzaPrestito

6.3.3 Amministratore

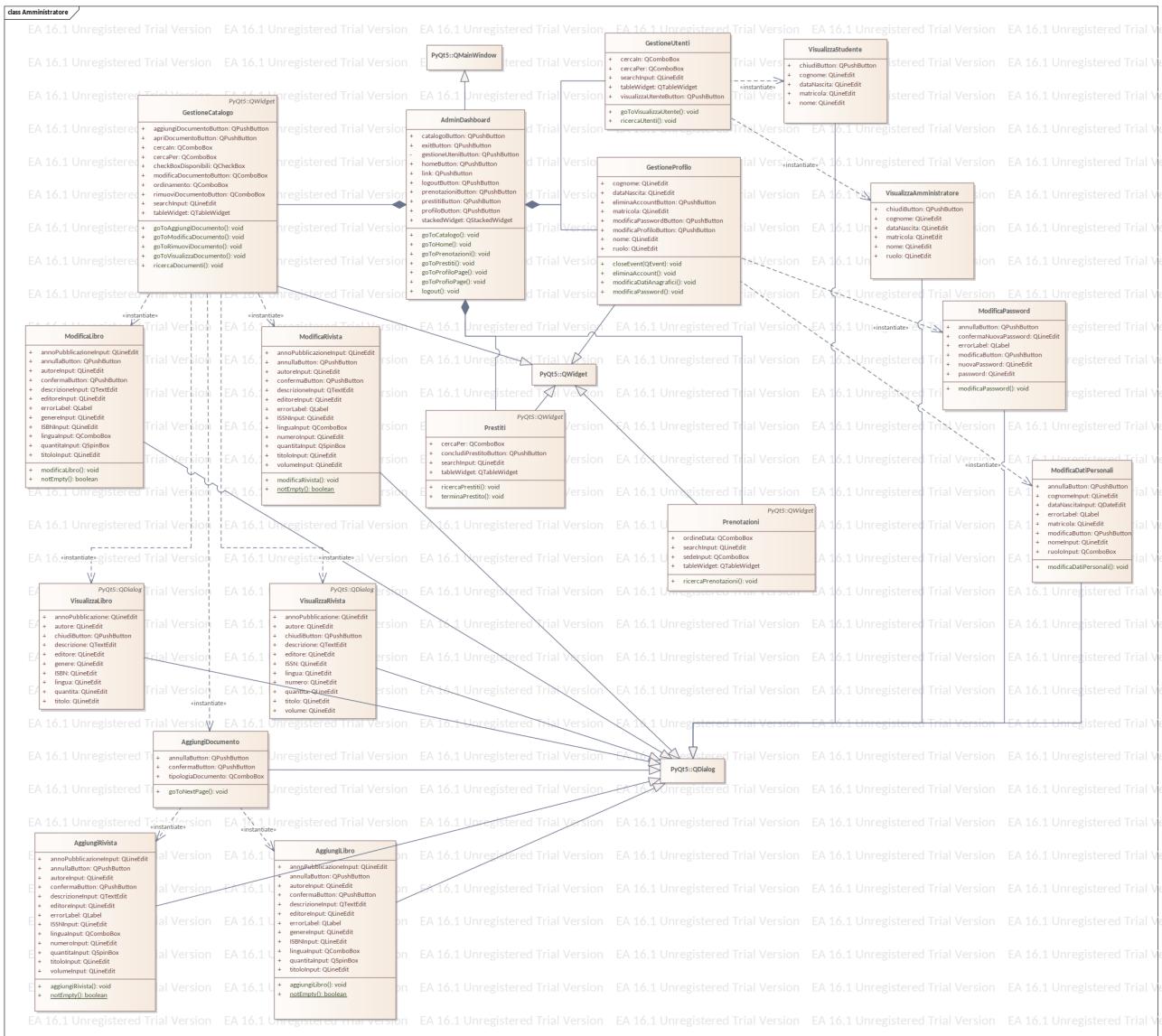


Figura 6.18: Amministratore

Come mostrato in 6.18 il package *Amministratore* include le seguenti classi ognuna delle quali importa un file .ui (User Interface):

- AdminDashboard
- GestioneCatalogo
- Prenotazioni
- Prestiti
- ModificaDatiPersonali
- ModificaPassword
- GestioneProfilo
- GestioneUtenti
- AggiungiDocumento
- AggiungiLibro
- AggiungiRivista
- VisualizzaLibro
- VisualizzaRivista
- ModificaLibro
- ModificaRivista
- VisualizzaAmministratore
- VisualizzaStudente

AdminDashboard



Figura 6.19: AdminDashboard

GestioneCatalogo

Catalogo

 search documents▼▼

Ordinamento: ▼ Solo disponibili

	Titolo	Autore	ID	Disponibilità
1	C e C++	Massimo ...	8849407424	3
2	Follia	Patrick McGrath	1234567891234	2
3	guida PHP	R. Nikolassy	8820334887	4
4	spider-man	stan lee	12345678	1

 Apri Aggiungi Modifica Rimuovi

◀ ▶

Figura 6.20: GestioneCatalogo

Prenotazioni

Gestione prenotazioni

 ricerca matricola studente

Sede:

Data:

	Matricola	Data	Ora arrivo	Ora andata	Sede
1	123456789	17/10/2023	08:30	09:30	Biblioteca ...

◀ ▶

Figura 6.21: Prenotazioni

Prestiti

Gestione prestiti

 ricerca prestitiCodice 

	Codice	Matricola student	ID documento	Scadenza
1	3	123456789	8849407424	16/11/2023
2	4	123456789	12345678	16/11/2023

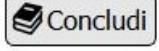

 

Figura 6.22: Prestiti

ModificaDatiPersonali

Modifica dati personali ? ×

Matricola:

Nome:

Cognome:

Data di nascita: ▾

Ruolo ▾

Figura 6.23: ModificaDatiPersonali

ModificaPassword

The screenshot shows a window titled "Cambia password" with a logo of a stylized "S" icon. In the top right corner are buttons for "?" and "X". The main area contains three input fields: "Password precedente:" with a blue border, "Nuova password:" with a white border, and "Conferma nuova password:" with a white border. Below these is a row of two buttons: "Modifica" with a pencil icon and "Annulla" with a red X icon.

Cambia password	?	X
Password precedente:	<input type="text"/>	
Nuova password:	<input type="text"/>	
Conferma nuova password:	<input type="text"/>	
<input type="button" value="Modifica"/>		<input type="button" value="Annulla"/>

Figura 6.24: ModificaPassword

GestioneProfilo

Informazioni personali

Matricola:

Nome:

Cognome:

Data di nascita:

Ruolo:

[Modifica dati personali](#) [Cambia password](#) [Elimina account](#)

Figura 6.25: GestioneProfilo

GestioneUtenti

Gestione Utenti

🔍 search users Matricola ▼ Tutto ▼

	Matricola	Nome	Cognome	Tipolo
1	123456789	studente	studente	Studente
2	987654321	admin	admin	Amministra

🕒 Visualizza

◀ ▶

Figura 6.26: GestioneUtenti

AggiungiDocumento

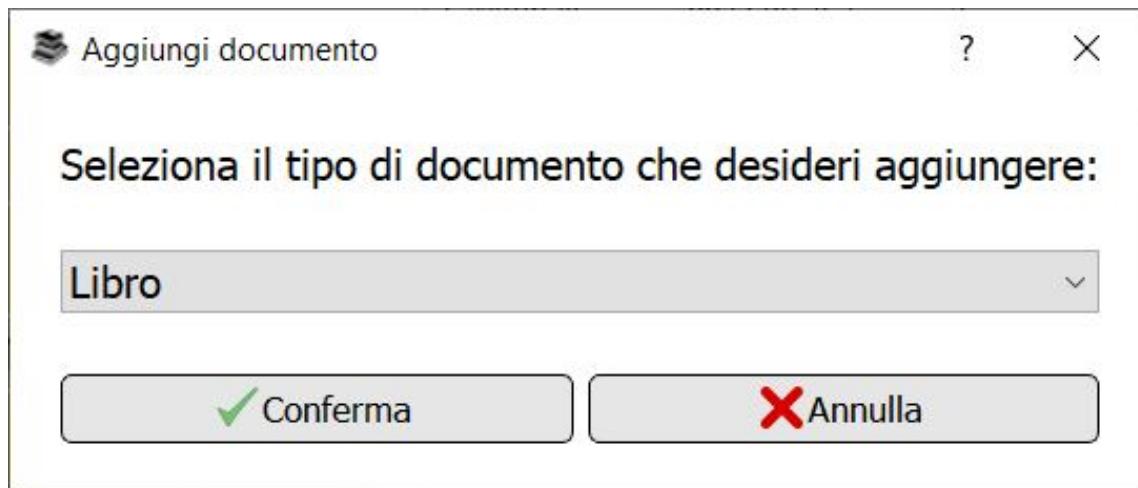


Figura 6.27: AggiungiDocumento

AggiungiLibro

Aggiungi-modifica libro ? ×

Titolo:

Autore:

Editore:

lingua: italiano ▾

Anno pubblicazione:

Genere:

ISBN:

Disponibilità: 0 unità ▾

Descrizione:

Conferma Annulla

Figura 6.28: AggiungiLibro

AggiungiRivista

Aggiungi-modifica rivista ? ×

Titolo:

Autore:

Editore:

lingua:

Anno pubblicazione:

ISSN:

Volume:

Numero:

Disponibilità: ▲▼

Descrizione:

 Conferma  Annulla

Figura 6.29: AggiungiRivista

ModificaLibro

Aggiungi-modifica libro ? ×

Titolo:	guida PHP
Autore:	R. Nikolassy
Editore:	hoepli
lingua:	italiano ▾
Anno pubblicazione:	2005
Genere:	informatica
ISBN:	8820334887
Disponibilità:	4 unità ▾
Descrizione:	Consigliato ai principianti

 Conferma  Annulla

Figura 6.30: ModificaLibro

ModificaRivista

Aggiungi-modifica rivista ? ×

Titolo:	spider-man
Autore:	stan lee
Editore:	marvel
lingua:	italiano ▾
Anno pubblicazione:	1998
ISSN:	12345678
Volume:	1
Numero:	3
Disponibilità:	1 unità ▾
Descrizione:	Rivista anno 2009

Conferma Annulla

Figura 6.31: ModificaRivista

VisualizzaLibro

 Visualizza libro ? ×

Titolo:	guida PHP
Autore:	R. Nikolassy
Editore:	hoepli
lingua:	italiano
Anno pubblicazione:	2005
Genere:	informatica
ISBN:	8820334887
Disponibilità:	4
Descrizione:	Consigliato ai principianti

Chiudi

Figura 6.32: VisualizzaLibro

VisualizzaRivista

 Visualizza rivista ? X

Titolo:	spider-man
Autore:	stan lee
Editore:	marvel
lingua:	italiano
Anno pubblicazione:	1998
ISSN:	12345678
Volume:	1
Numero:	3
Disponibilità:	1
Descrizione:	Rivista anno 2009

Chiudi

Figura 6.33: VisualizzaRivista

VisualizzaAmministratore

 Visualizza amministratore	?	X
Matricola:	987654321	
Nome:	admin	
Cognome:	admin	
Data di nascita:	01/01/70	
ruolo:	Bibliotecario/a	
Chiudi		

Figura 6.34: VisualizzaAmministratore

VisualizzaStudente

 Visualizza studente	?	X
Matricola:	123456789	
Nome:	studente	
Cognome:	studente	
Data di nascita:	01/01/00	
Stato:	Account attivo	
<input type="button" value="Chiudi"/>		

Figura 6.35: VisualizzaStudente

7. DIAGRAMMI DI SEQUENZA

Per avere una maggiore chiarezza del funzionamento del software sono stati realizzati dei diagrammi di sequenza. I casi d'uso analizzati sono i seguenti:

- AggiungiDocumento
- ModificaDocumento
- RimuoviDocumento
- Autenticazione
- EffettuaPrestito
- VisualizzaListaPrenotazioni
- VisualizzaPrestito

7.1 AggiungiDocumento

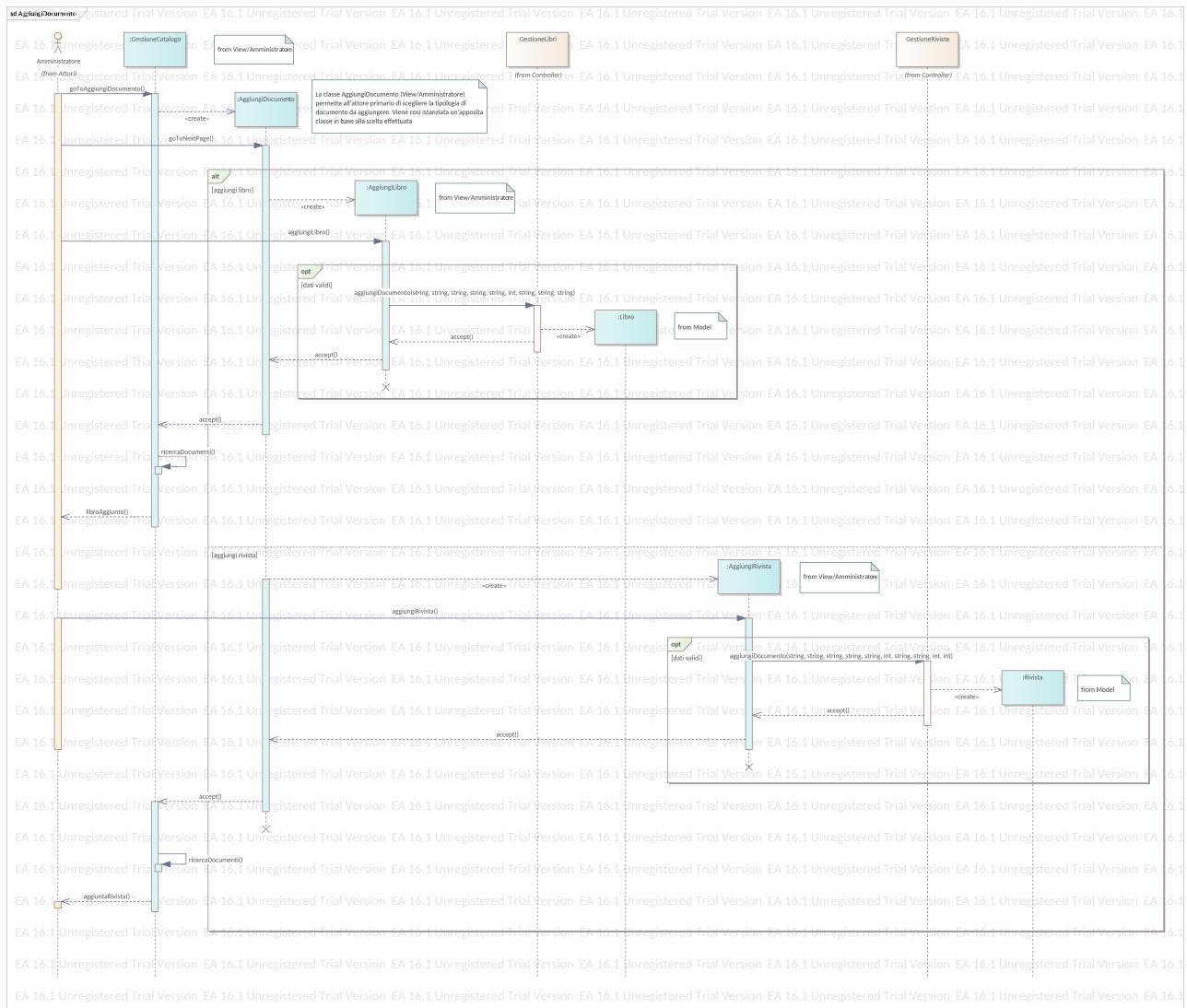


Figura 7.1: sd: AggiungiDocumento

7.2 ModificaDocumento

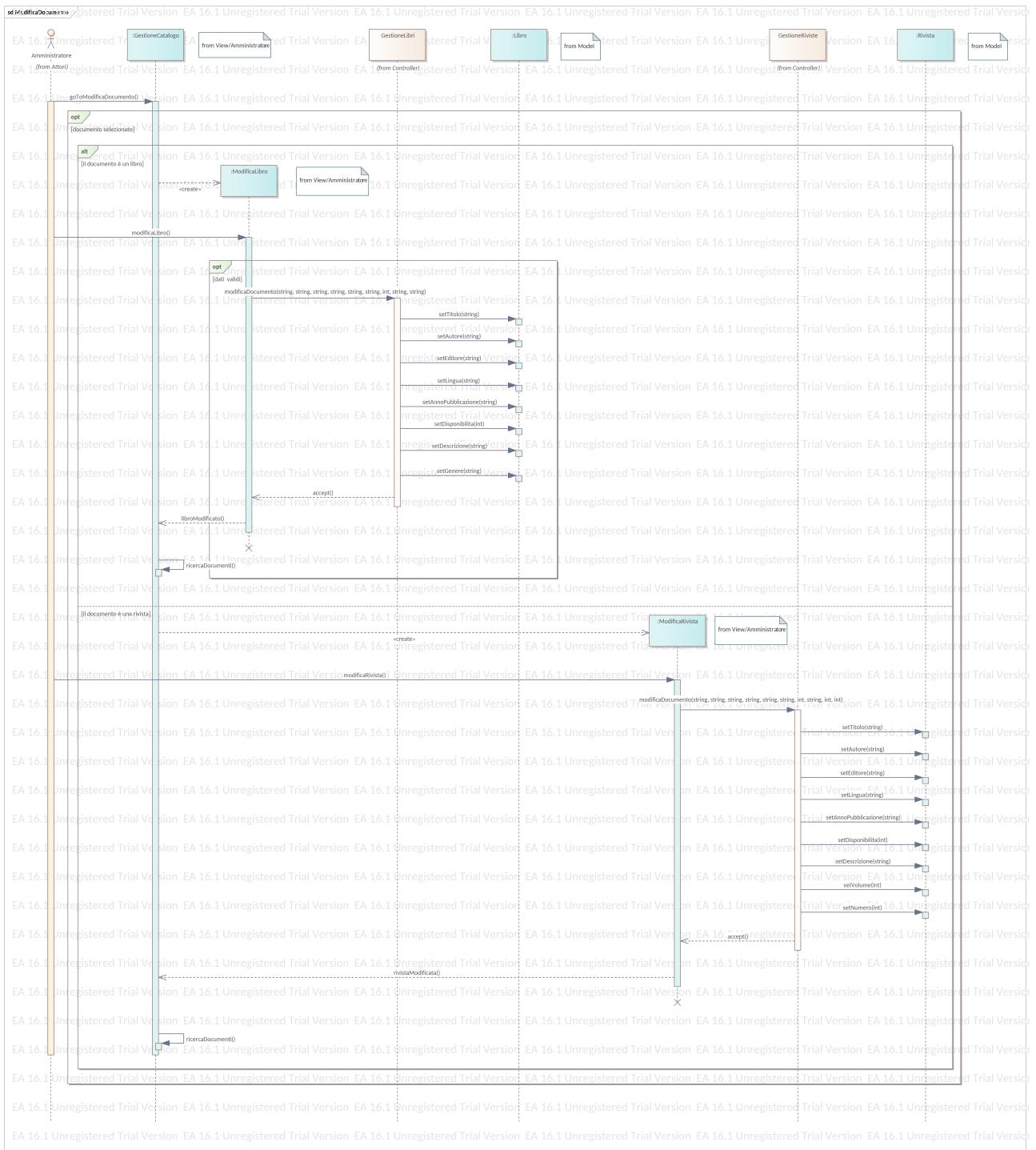


Figura 7.2: sd: ModificaDocumento

7.3 RimuoviDocumento

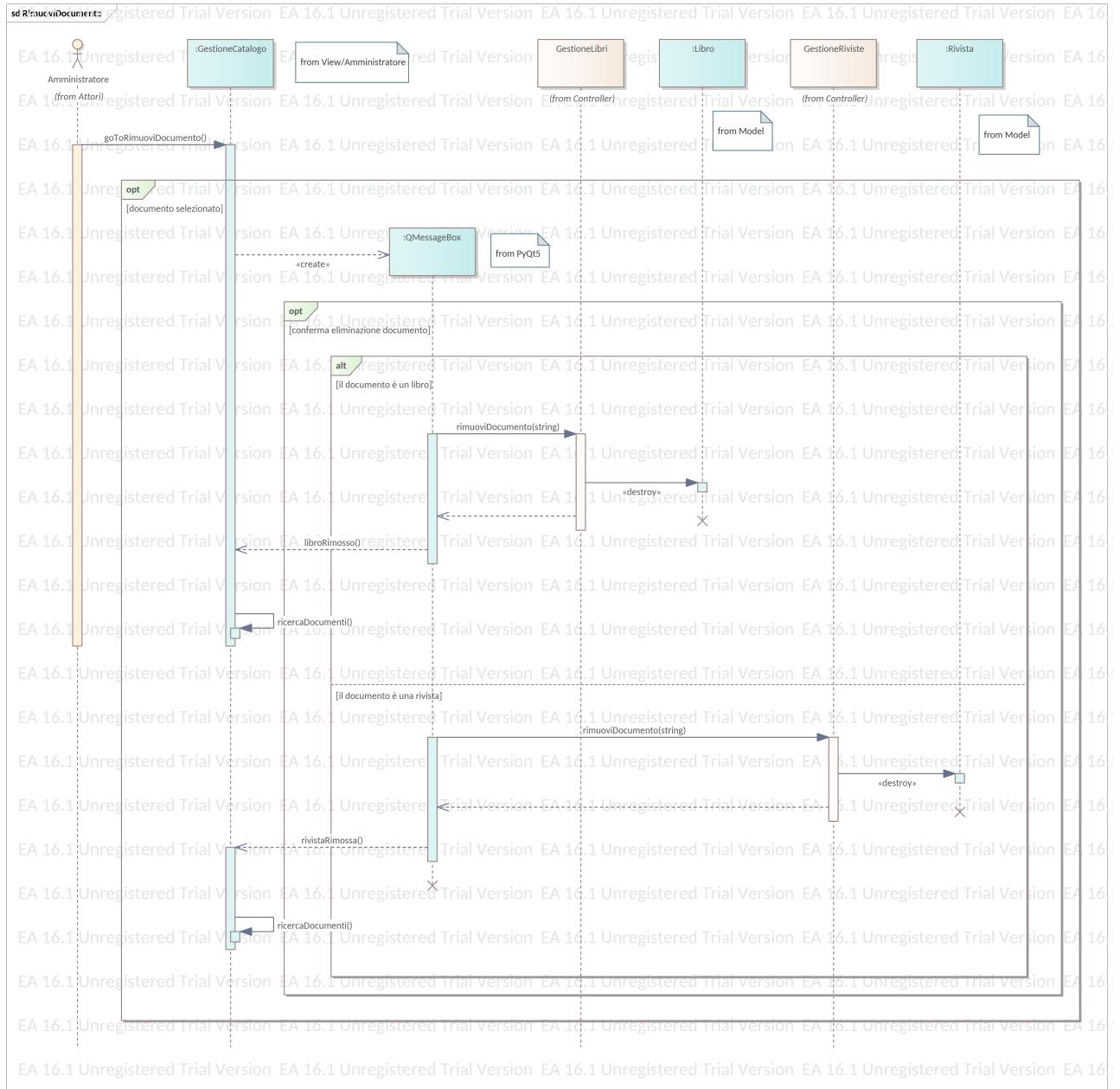


Figura 7.3: sd: RimuoviDocumento

7.4 Autenticazione

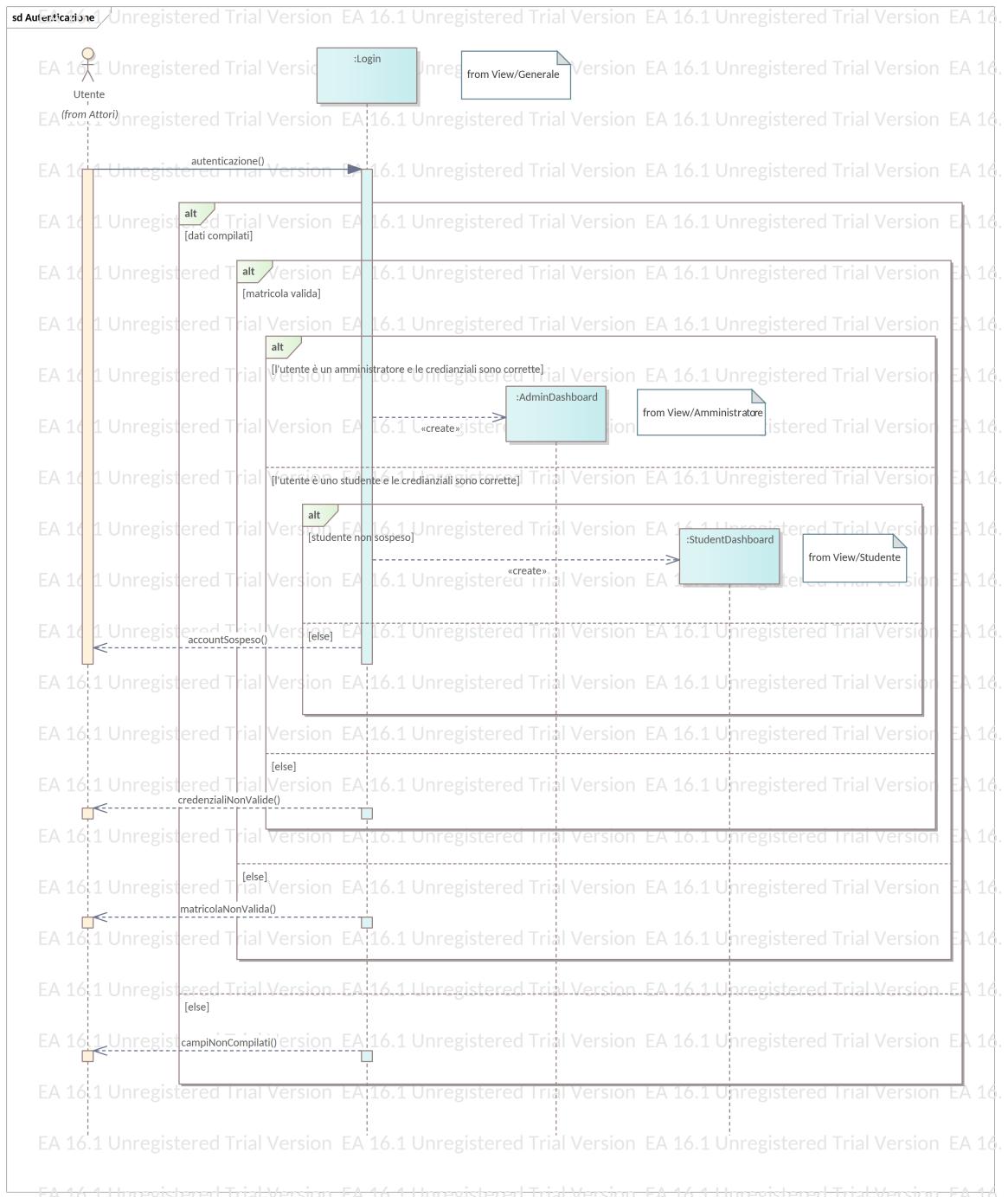


Figura 7.4: sd: Autenticazione

7.5 EffettuaPrestito

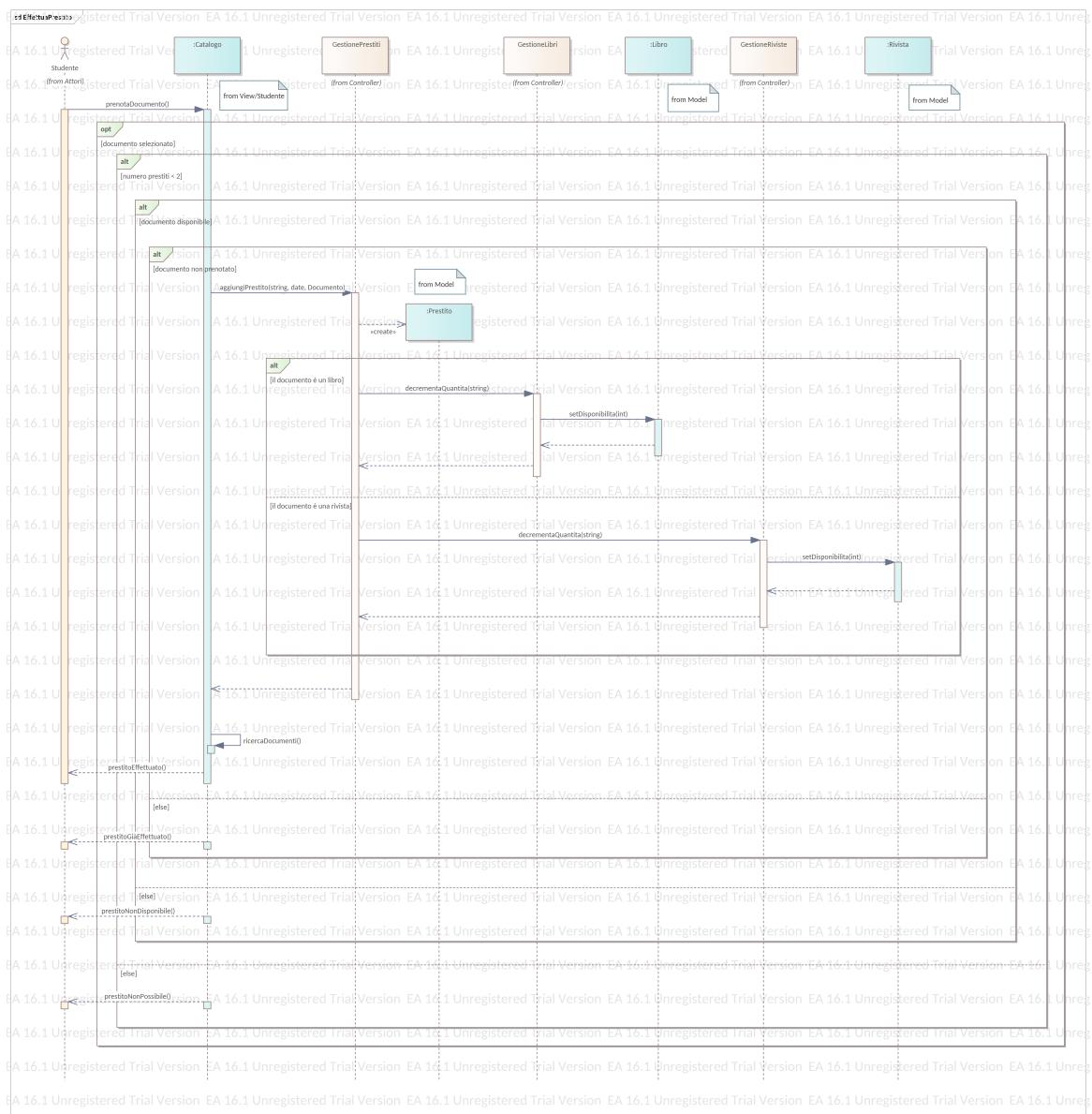


Figura 7.5: sd: EffettuaPrestito

7.6 VisualizzaListaPrenotazioni

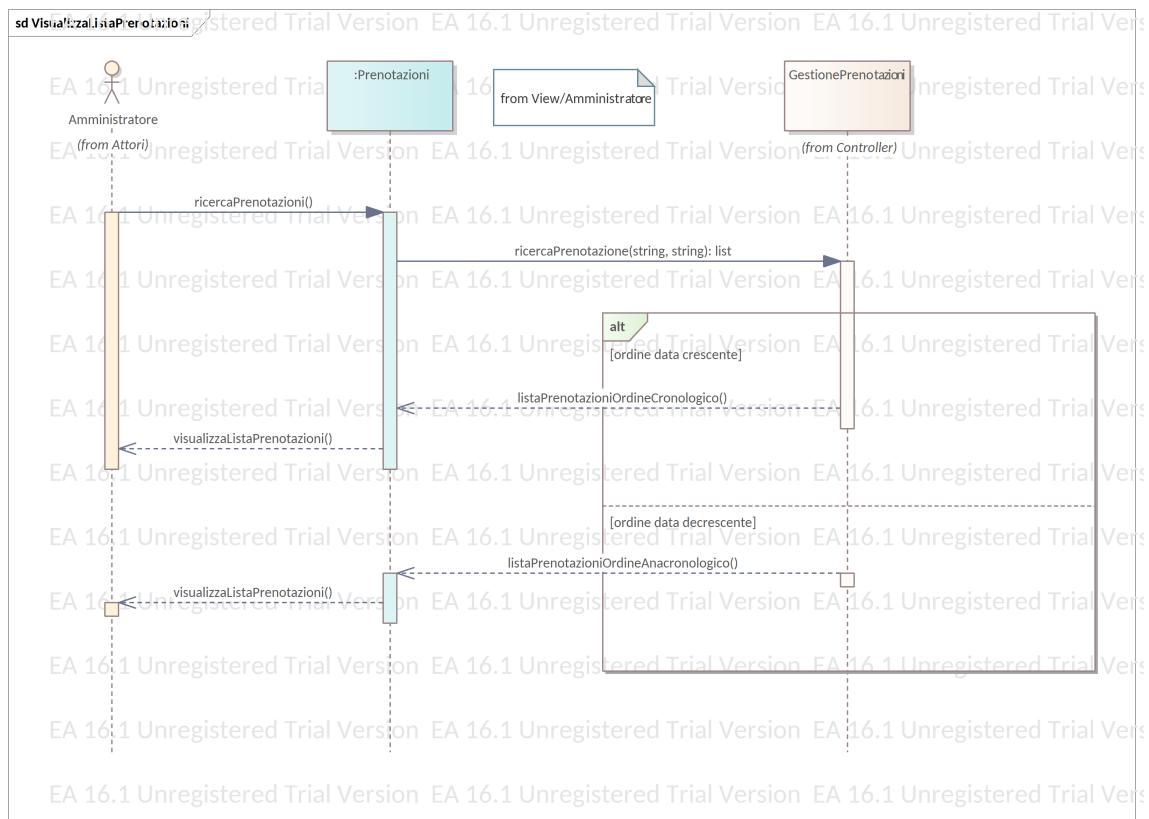


Figura 7.6: sd: VisualizzaListaPrenotazioni

7.7 VisualizzaPrestito

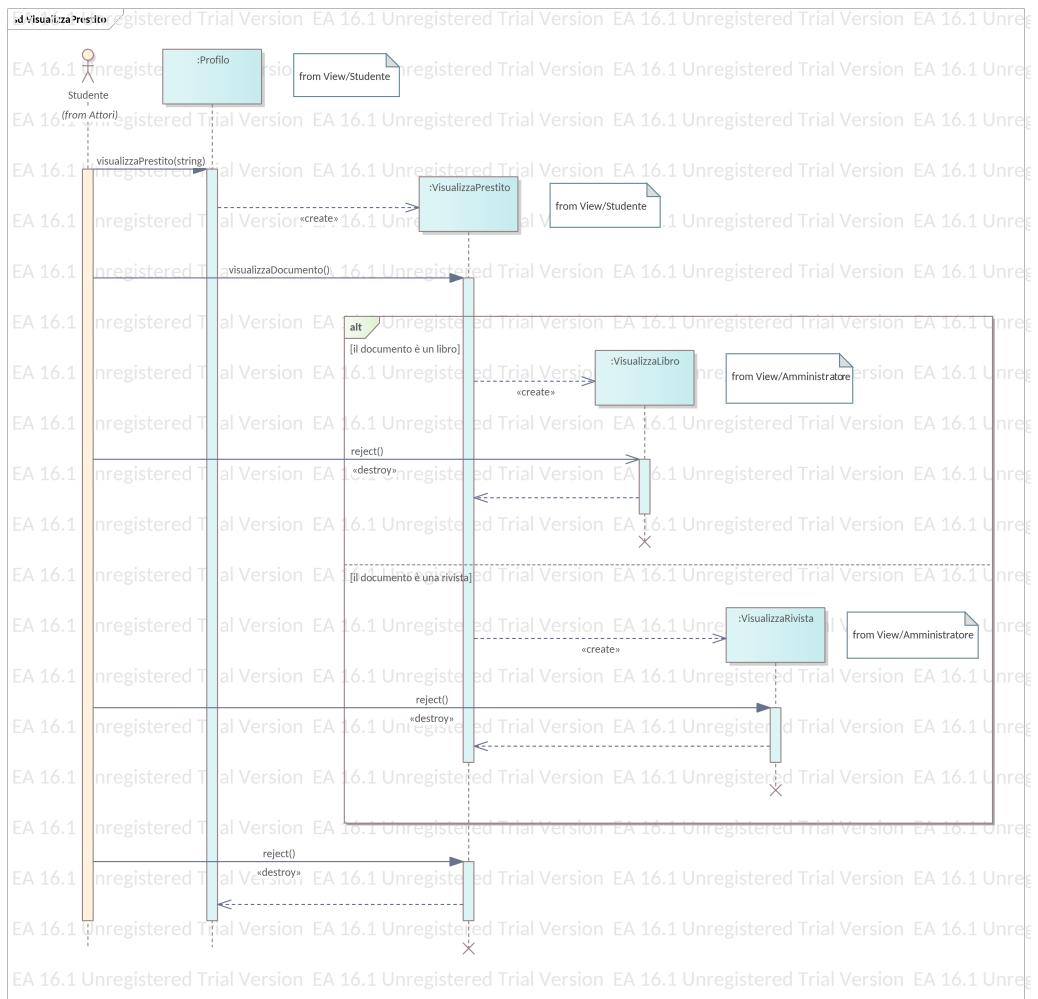


Figura 7.7: sd: VisualizzaPrestito

8. DIAGRAMMI DI ATTIVITÀ

Per avere una maggiore chiarezza del funzionamento del software sono stati realizzati dei diagrammi di attività. I casi d'uso analizzati sono i seguenti:

- Autenticazione
- TerminaPrestito
- VisualizzaListaPrenotazioni
- RicercaPrenotazione

8.1 Autenticazione

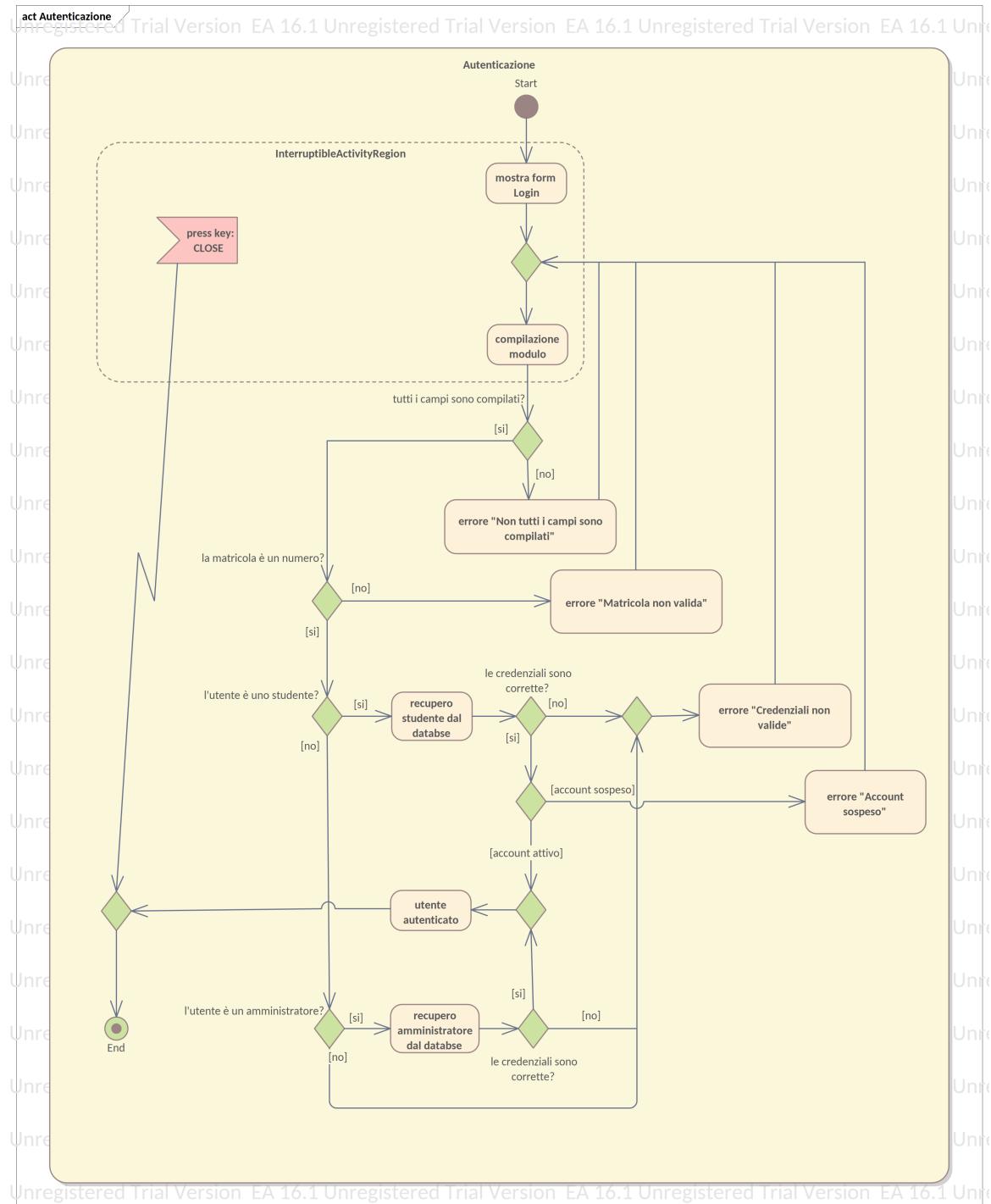


Figura 8.1: act: Autenticazione

8.2 VisualizzaListaPrenotazioni

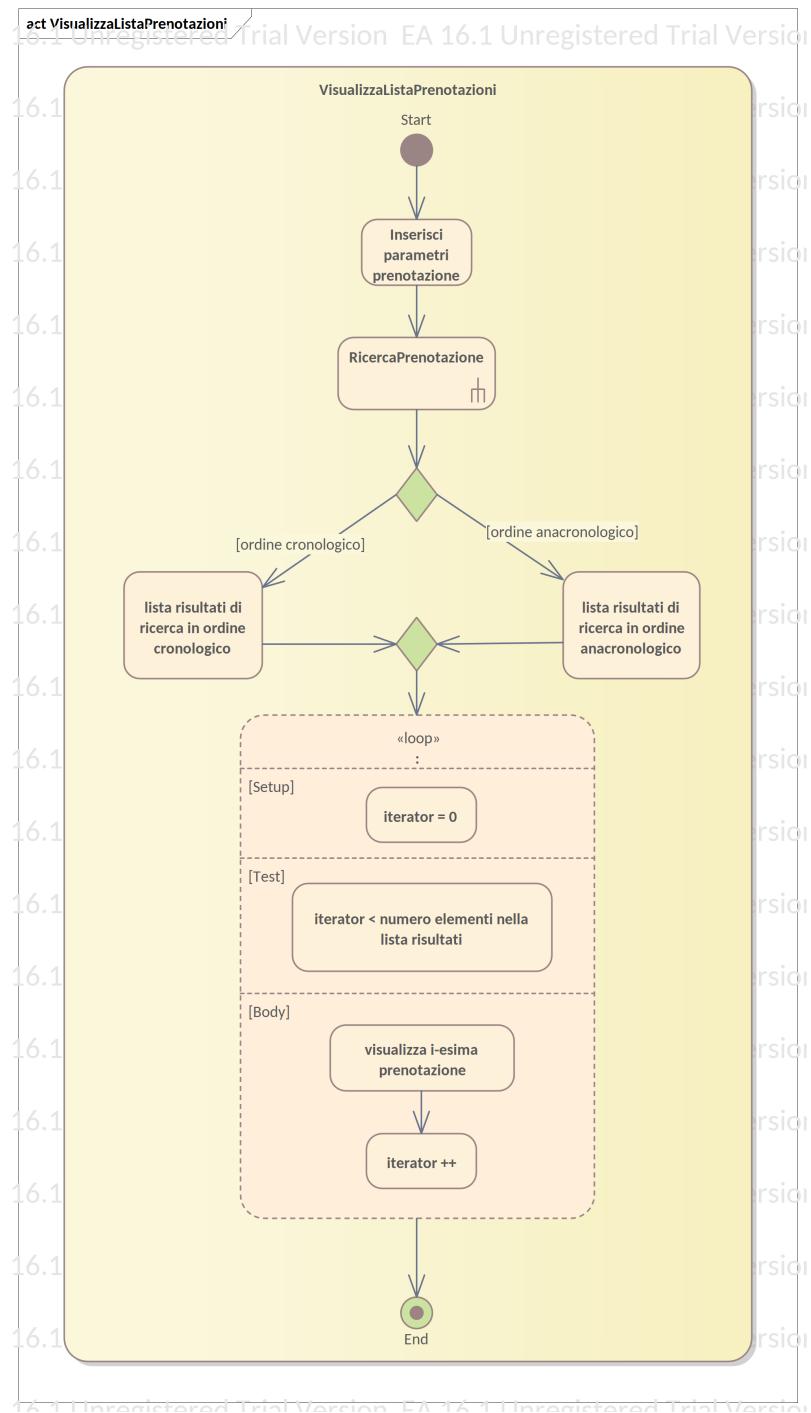


Figura 8.2: act: VisualizzaListaPrenotazioni

8.3 act-RicercaPrenotazione

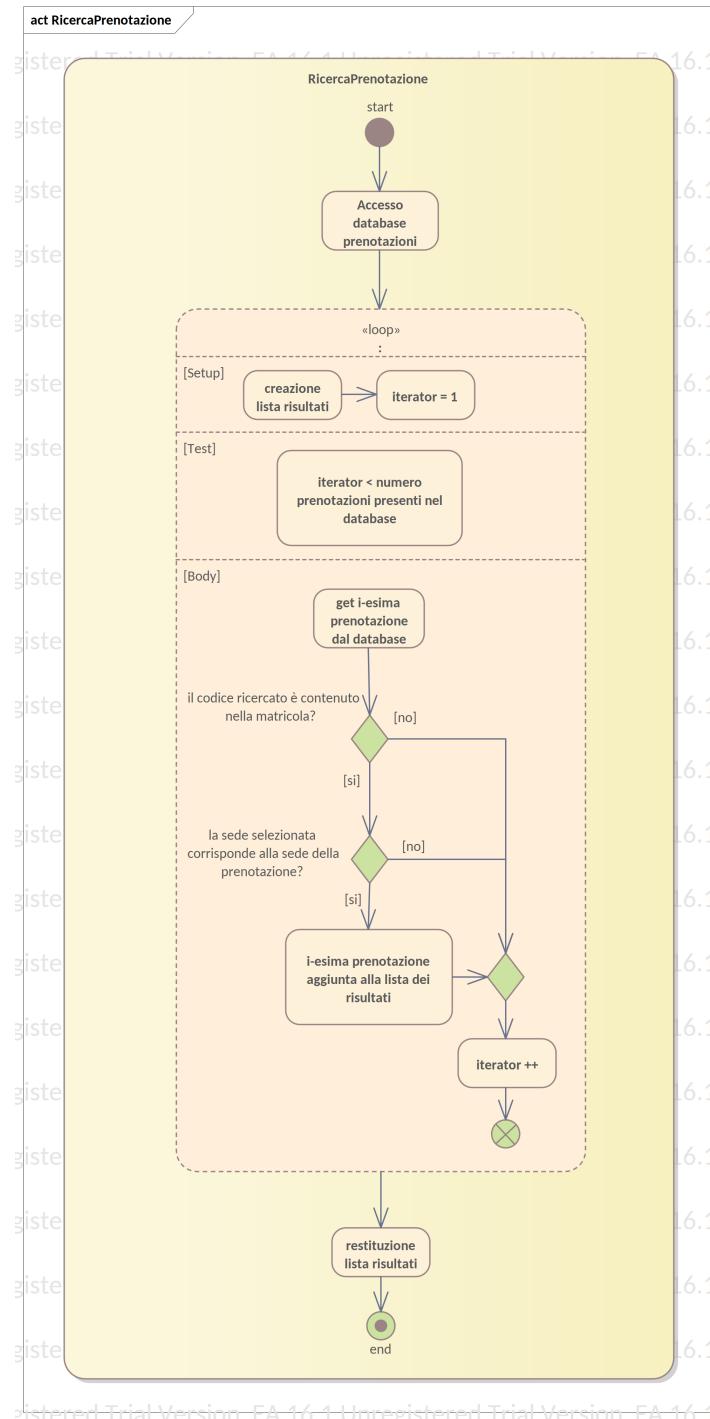


Figura 8.3: act: RicercaPrenotazione

8.4 TerminaPrestito

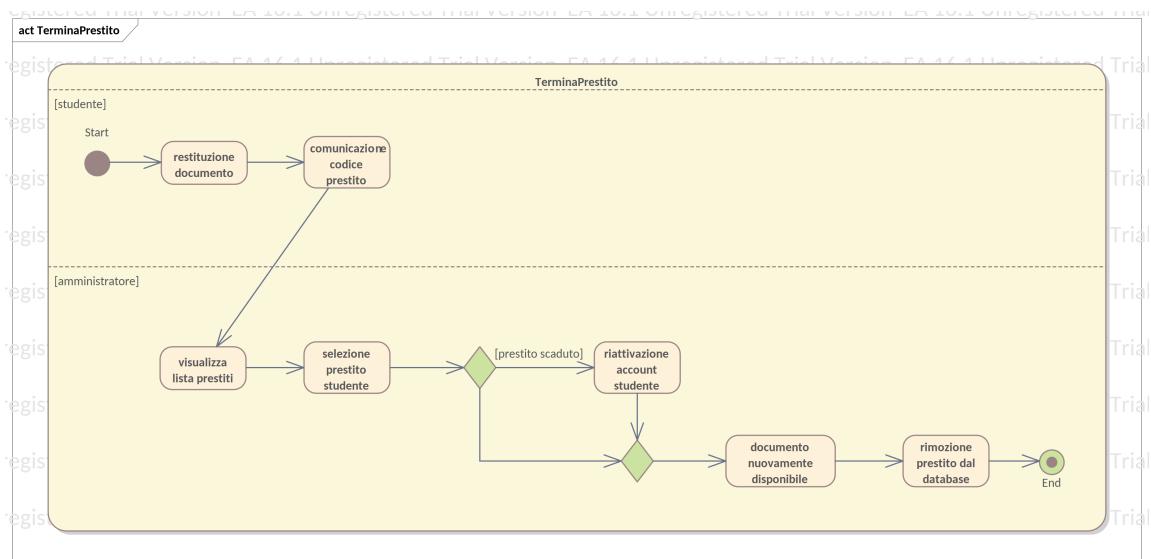


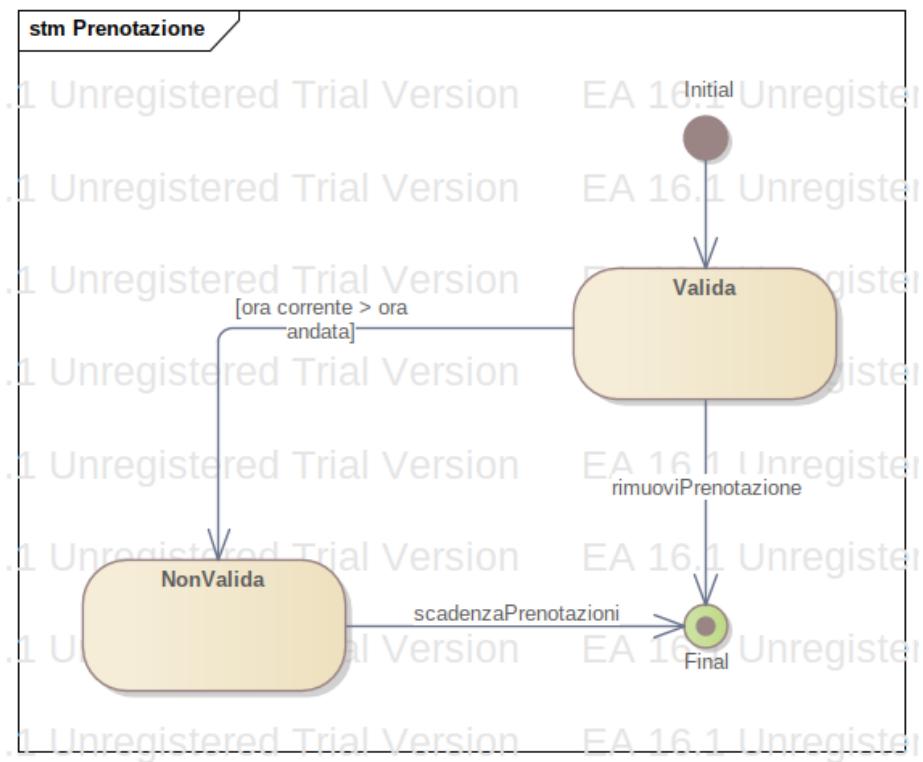
Figura 8.4: act: TerminaPrestito

9. DIAGRAMMI DELLE MACCHINE A STATI

Il diagramma delle macchine a stati raffigura gli oggetti di una classe delineando i vari stati ad essi collegati all'interno del sistema: Le classi per cui si è realizzato tale diagramma sono:

- Prenotazione
- Prestito
- Studente

9.1 Prenotazione



9.2 Prestito

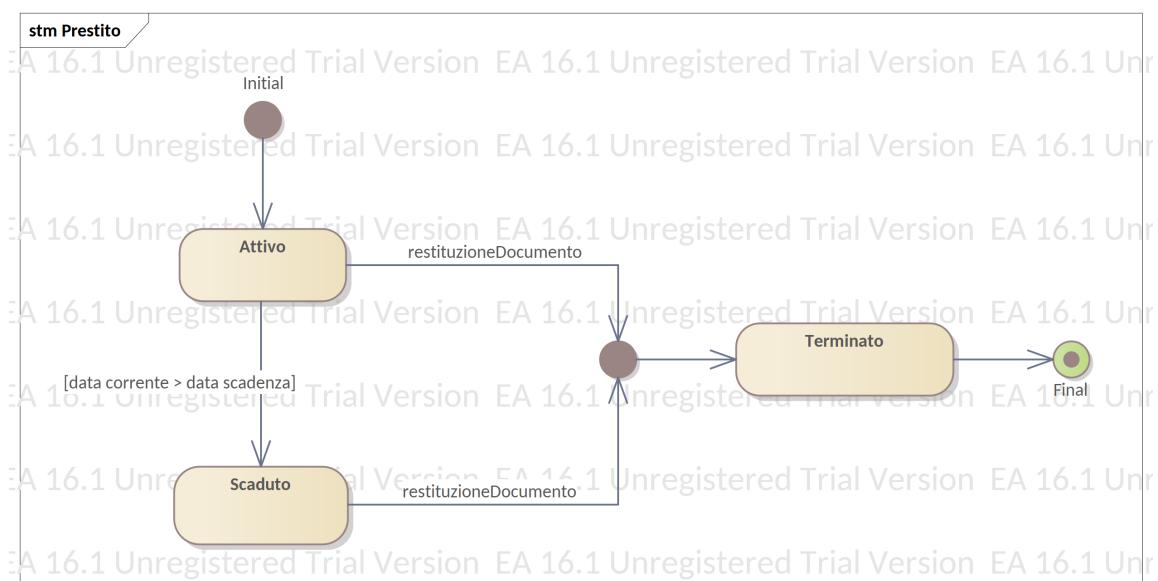


Figura 9.2: Diagramma delle macchine a stati: Prestito

9.3 Studente

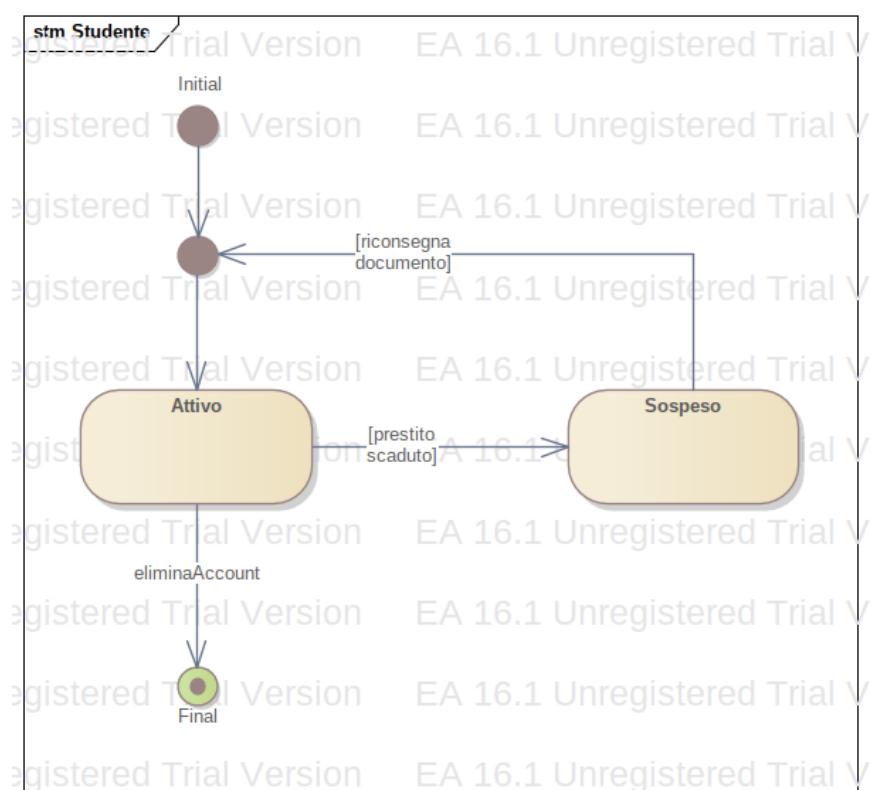


Figura 9.3: Diagramma delle macchine a stati: Studente

10. DIAGRAMMA DEI COMONENTI

Il diagramma dei componenti rappresentato in 10.1 mostra la struttura del sistema software dal punto di vista dello studente.

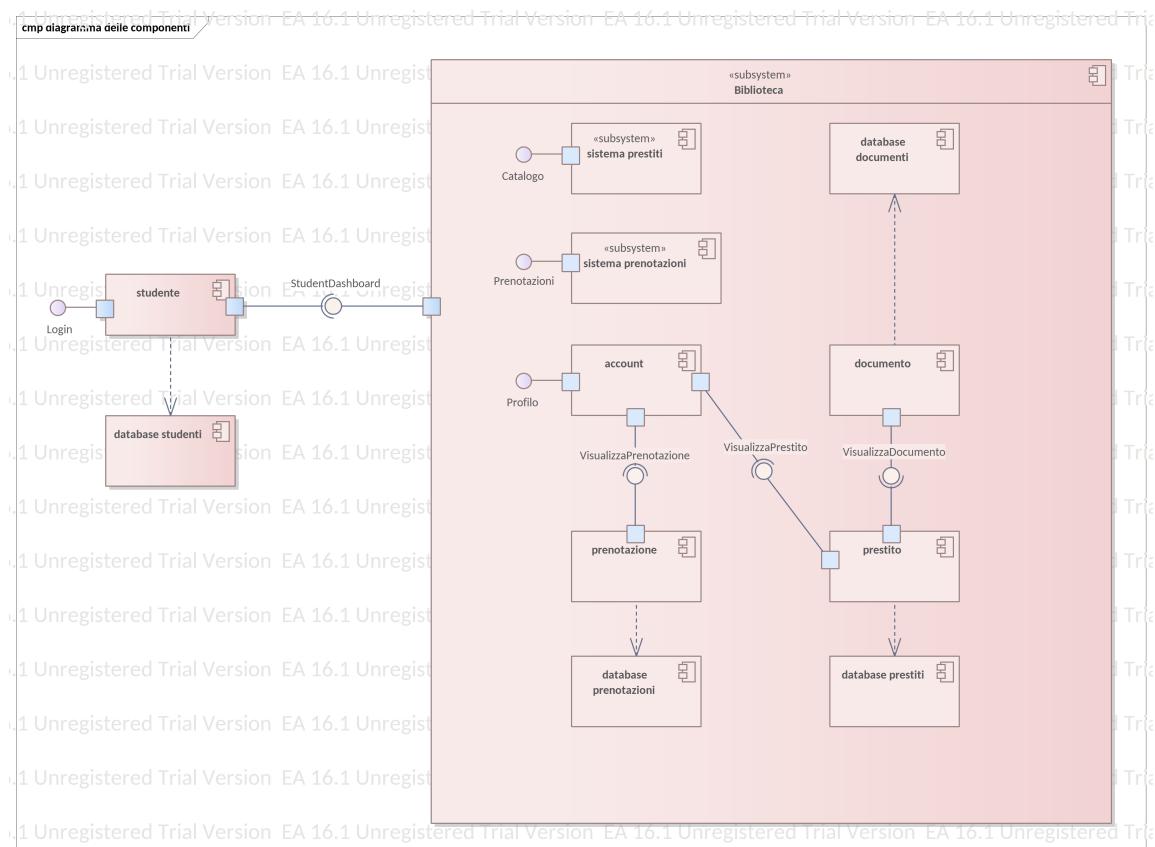


Figura 10.1: Diagramma dei componenti

11. DIAGRAMMA DI DEPLOYMENT

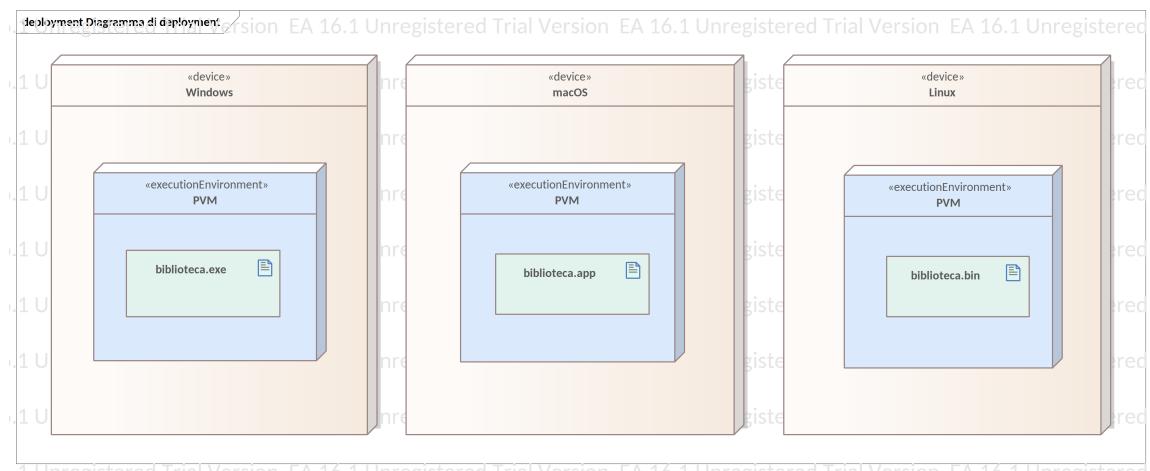


Figura 11.1: Diagramma di deployment

12. UNIT TEST

Durante la fase di implementazione, tramite il modulo **unittest** si sono utilizzati segmenti di codice per testare il corretto funzionamento del software. La classe "test" contenuta nella cartella "Tests" situata nella directory principale del progetto contiene le seguenti funzioni:

- test_studente
- test_amministratore
- test_libro
- test_rivista
- test_prestito
- test_prenotazione

12.1 test_studente

```
def test_studente(self):
    #aggiunta studente
    GestioneStudenti.aggiungiUtente('Kristian', 'Likaj', '1097605', '12345678', datetime.date(2000, 10, 10))
    self.assertIsInstance(GestioneStudenti.getUtente('1097605'), Studente)

    #verifica modifiche studente
    GestioneStudenti.modificaUtente('Kristian', 'Likaj', '1097605', 'password', datetime.date(2000, 10, 10))
    self.assertEqual(GestioneStudenti.getUtente("1097605").getPassword(), "password")

    #rimozione studente
    GestioneStudenti.rimuoviUtente("1097605")
    self.assertIsNone(GestioneStudenti.getUtente("1097605"))
```

Figura 12.1: test Studente

12.2 test_amministratore

```
def test_amministratore(self):
    # aggiunta amministratore
    GestioneAmministratori.aggiungiUtente('Mario', 'Rossi', '12345678', 'asdfghjkl', datetime.date(1970, 11, 11),
                                             'Assistente')
    self.assertIsInstance(GestioneAmministratori.getUtente('12345678'), Amministratore)

    # verifica modifiche amministratore
    GestioneAmministratori.modificaUtente('Mario', 'Rossi', '12345678', 'password', datetime.date(1970, 11, 11),
                                             'Assistente')
    self.assertEqual(GestioneAmministratori.getUtente("12345678").getPassword(), "password")

    # rimozione amministratore
    GestioneAmministratori.rimuoviUtente("12345678")
    self.assertIsNone(GestioneAmministratori.getUtente("12345678"))
```

Figura 12.2: test Amministratore

12.3 test_libro

```
def test_libro(self):
    #aggiunta libro
    GestioneLibri.aggiungiDocumento('Follia', 'Patrick McGrath', 'Adelphi', 'italiano', '2012', 2, 'bellissimo',
                                    '1234567891234', 'Romanzo')
    self.assertIsInstance(GestioneLibri.getDocumento('1234567891234'), Libro)

    #verifica modifica libro
    GestioneLibri.modificaDocumento('1234567891234', 'Follia', 'Patrick McGrath', 'Adelphi', 'italiano', '2012',
                                    2, 'consigliato', 'Romanzo')
    self.assertEqual(GestioneLibri.getDocumento('1234567891234').getDescrizione(), "consigliato")

    #verifica incremento quantità
    quantitaIniziale = GestioneLibri.getDocumento("1234567891234").getDisponibilita()
    GestioneLibri.incrementaQuantita("1234567891234")
    self.assertEqual(GestioneLibri.getDocumento('1234567891234').getDisponibilita(), quantitaIniziale + 1)

    # verifica decremento quantità
    GestioneLibri.decrementaQuantita("1234567891234")
    self.assertEqual(GestioneLibri.getDocumento('1234567891234').getDisponibilita(), quantitaIniziale)

    #rimozione libro
    GestioneLibri.rimuoviDocumento("1234567891234")
    self.assertIsNone(GestioneLibri.getDocumento('1234567891234'))
```

Figura 12.3: test Libro

12.4 test_rivista

```
def test_rivista(self):
    # aggiunta rivista
    GestioneRiviste.aggiungiDocumento('Follia', 'Patrick McGrath', 'Adelphi', 'italiano', '2012', 2, 'bellissimo',
                                      '12341234', 1, 3)
    self.assertIsInstance(GestioneRiviste.getDocumento('12341234'), Rivista)

    # verifica modifica rivista
    GestioneRiviste.modificaDocumento('12341234', 'Follia', 'Patrick McGrath', 'Adelphi', 'italiano', '2012', 2,
                                      'consigliato', 1, 3)
    self.assertEqual(GestioneRiviste.getDocumento('12341234').getDescrizione(), "consigliato")

    # verifica incremento quantità
    quantitaIniziale = GestioneRiviste.getDocumento("12341234").getDisponibilita()
    GestioneRiviste.incrementaQuantita("12341234")
    self.assertEqual(GestioneRiviste.getDocumento('12341234').getDisponibilita(), quantitaIniziale + 1)

    # verifica decremento quantità
    GestioneRiviste.decrementaQuantita("12341234")
    self.assertEqual(GestioneRiviste.getDocumento('12341234').getDisponibilita(), quantitaIniziale)

    # rimozione rivista
    GestioneRiviste.rimuoviDocumento("12341234")
    self.assertIsNone(GestioneRiviste.getDocumento('12341234'))
```

Figura 12.4: test Rivista

12.5 test_prestito

```
def test_prestito(self):
    #aggiungi prestito
    GestioneLibri.aggiungiDocumento('Follia', 'Patrick McGrath', 'Adelphi', 'italiano', '2012', 2, 'bellissimo',
                                     '1234567891234', 'Romanzo')
    GestioneStudenti.aggiungiUtente('Kristian', 'Likaj', '1097605', '12345678', datetime.date(2000, 10, 10))
    libro = GestioneLibri.getDocumento('1234567891234')
    dataInizio = datetime.date.today()
    dataScadenza = dataInizio + datetime.timedelta(days=30)
    GestionePrestiti.aggiungiPrestito('1097605', dataInizio, libro)
    prestiti = GestionePrestiti.ricercaPrestitoPerMatricola('1097605')
    self.assertIsInstance(prestiti[0], Prestito)

    #modifica prestito
    GestionePrestiti.prorogaScadenzaPrestito(prestiti[0].getCodice())
    self.assertEqual(prestiti[0].getDataScadenza(), dataScadenza)

    #rimozione prestito
    GestionePrestiti.rimuoviPrestito(prestiti[0].getCodice())
    self.assertIsNone(GestionePrestiti.getPrestito(prestiti[0].getCodice()))
```

Figura 12.5: test Prestito

12.6 test_prenotazione

```
def test_prenotazione(self):
    #aggiunta prenotazione
    GestioneStudenti.aggiungiUtente('Kristian', 'Likaj', '1097605', '12345678', datetime.date(2000, 10, 10))
    locale.setlocale(locale.LC_ALL, "it")
    adesso = datetime.datetime.now()
    oraInizio = datetime.time(hour=adesso.hour, minute=adesso.minute, second=adesso.second)
    oraFine = adesso + datetime.timedelta(minutes=30)
    GestionePrenotazioni.aggiungiPrenotazione("1097605", "Biblioteca Tecnico-Scientifica-Biomedica",
                                                datetime.date.today(), oraInizio, oraFine)
    self.assertIsInstance(GestionePrenotazioni.getPrenotazione("1097605"), Prenotazione)

    #rimozione prenotazione
    GestionePrenotazioni.rimuoviPrenotazione("1097605")
    self.assertIsNone(GestionePrenotazioni.getPrenotazione("1097605"))
    GestioneStudenti.rimuoviUtente("1097605")
```

Figura 12.6: test Prenotazione

13. NOTE

Il codice sorgente del software è disponibili sulla piattaforma GitHub al seguente indirizzo <https://github.com/FrancescoFinizii/Biblioteca-universitaria.git>. Il software è stato testato sia sul sistema operativo Windows che Linux. Per maggiori dettagli si consiglia di visionare il file README.md situato nella repository del progetto.