



Lazioteca - Documento di Design

Sebastiano Deodati 2025953

Revisione 2024-04-30

Indice

1 Cronologia delle revisioni	4
2 Design architetturale	4
3 Classi di Design	4
3.1 Comunicazione con la base dati e ORM	4
3.2 Suddivisione delle classi in package	5
3.3 Package it.deodatisw.lazioteca.datamodel	5
3.4 Package it.deodatisw.lazioteca.ui	6
3.5 Package it.deodatisw.lazioteca.comunicazione	6
3.6 Package it.deodatisw.lazioteca.gestione_account	7
3.7 Package it.deodatisw.lazioteca.vetrina	7
3.8 Package it.deodatisw.lazioteca.fruizione_risorse	8
3.9 Package it.deodatisw.lazioteca.gestione_biblioteche	9
4 Diagrammi di sequenza di Design	9
4.1 GestioneAccount	10
4.1.1 SDD01	10
4.1.2 SDD02	11
4.1.3 SDD22	11
4.2 Vetrina	12
4.2.1 SDD03	12
4.2.2 SDD04	12
4.2.3 SDD05	12
4.2.4 SDD06	13
4.2.5 SDD07	13
4.2.6 SDD08	14
4.2.7 SDD09	15
4.2.8 SDD28	15
4.3 FruizioneRisorse	16
4.3.1 SDD10	16
4.3.2 SDD11	16
4.4 Noleggi	17
4.4.1 SDD12	17
4.4.2 SDD13	17
4.4.3 SDD14	18
4.4.4 SDD15	18
4.4.5 SDD26	18
4.4.6 SDD27	19

4.4.7	SDD29	19
4.5	GestioneBiblioteche	19
4.5.1	SDD16	19
4.5.2	SDD17	20
4.5.3	SDD18	20
4.5.4	SDD19	21
4.5.5	SDD20	21
4.6	GestioneCatalogo	22
4.6.1	SDD23	22
4.6.2	SDD24	22
4.6.3	SDD25	22
4.6.4	SDD30	23

1 Cronologia delle revisioni

Revisione	Descrizione
2025-01-09	Prima stesura (UC01, UC02, UC03, UC04)
2025-01-22	Aggiornata documentazione per UC05, UC06, UC07, UC09, UC11
2025-02-05	Aggiornata documentazione per UC12, UC15, UC17
2025-02-19	Aggiornata documentazione per UC16, UC18, UC25
2025-03-05	Aggiornata documentazione per UC23, UC24
2025-03-19	Aggiornata documentazione per UC26, UC27, UC29
2025-04-02	Aggiornata documentazione per UC30, UC08, UC13
2025-04-16	Aggiornata documentazione per UC14, UC10, UC19, UC20
2025-04-30	Aggiornata documentazione per UC22, UC28

2 Design architetturale

Come già discusso nello studio di fattibilità, il sistema verrà implementato come un applicativo RESTful. Il sistema seguirà quindi un'architettura client-server, dove i client sono i terminali degli utenti che si interfacciano, tramite il frontend, al backend che funge da server. Sia il frontend che il backend verranno serviti da quattro i server (mediante container Docker), mentre un firewall si occuperà, oltre che della sicurezza, del load balancing. La base dati sarà centralizzata e hostata su un server dedicato. Su ogni server è configurato un sistema di backup incrementale che memorizza le snapshot sul NAS.

3 Classi di Design

3.1 Comunicazione con la base dati e ORM

La comunicazione con la base dati sarà gestita dalla libreria JDBC (package `java.sql`), che consente di standardizzare le modalità di connessione alla base dati e di interrogazione, pertanto l'ambiente di sviluppo dovrà essere patchato con <https://jdbc.postgresql.org>. Per implementare l'Object-Relational Mapping è stato scelto di applicare il pattern *Data Access Object*, in particolare:

- Ogni classe della business logic è implementata come classe DAO (`UtenteDAO`, `BibliotecaDAO...`) con una corrispondente tabella nella base dati (`Utenti`, `Biblioteche...`)

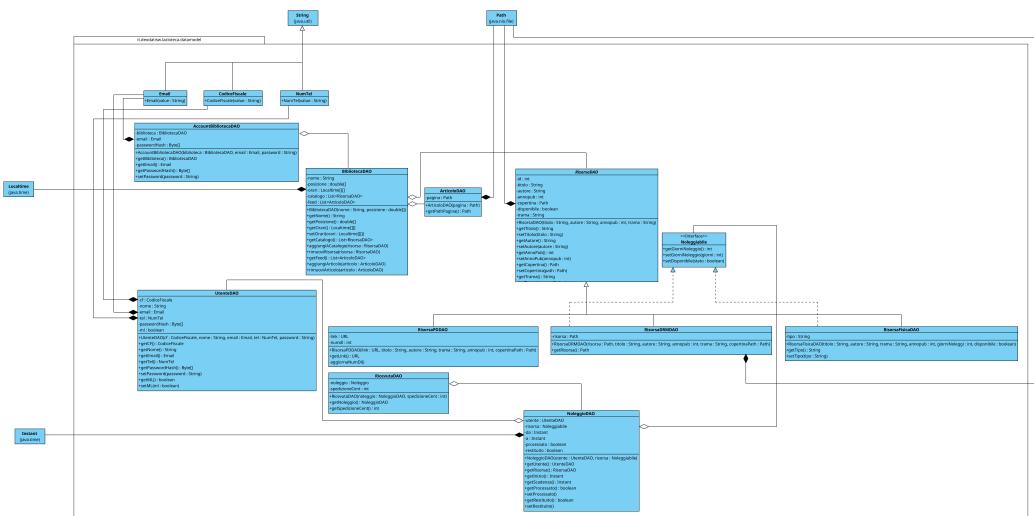
- L'invio di query alla base dati è gestito tramite l'interfaccia `java.sql.PreparedStatement`, per ogni query è aperta una nuova connessione apposita
 - Ogni costruttore implica una `INSERT INTO Class VALUES (?, ?, ...)`
 - Ogni getter implica una `SELECT attr FROM Class WHERE key = ?`
 - Ogni setter o comunque metodi che modificano lo stato di una entità implicano un `UPDATE Class SET attr = ? WHERE key = ?`

Ciò garantisce coerenza tra lo stato della business logic all'interno dell'applicazione e i dati memorizzati nella base dati in qualsiasi momento.

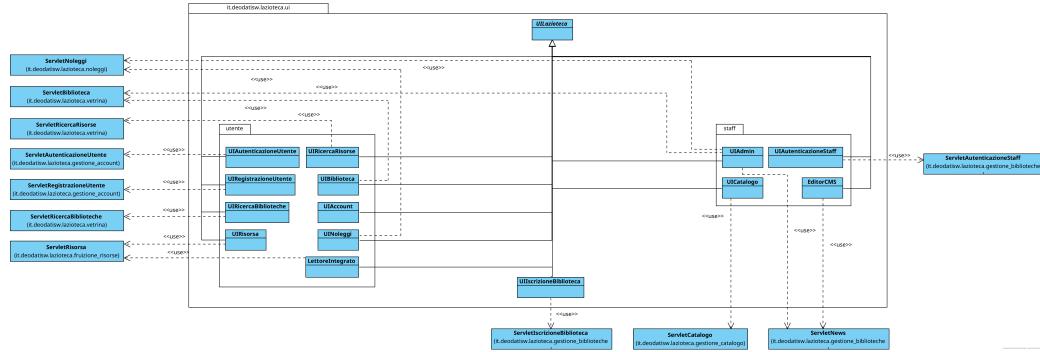
3.2 Suddivisione delle classi in package

Per la suddivisione delle classi, vista l'architettura RESTful da implementare, è stato scelto di applicare il pattern MVC (Model View Controller), secondo cui il progetto verrà suddiviso in tre componenti principali: modello logico dei dati (package datamodel), vista sui dati (package ui) e controllo interazione delle due componenti e dell'utente (altri package). Per riflettere la suddivisione in sottosistemi è stato scelto di suddividere il controller in un package per ogni sottosistema. È stato scelto inoltre di mettere le interfacce di comunicazione (email, SMS) in un package dedicato. I package sono rappresentati ognuno in un diagramma separato per favorire la leggibilità.

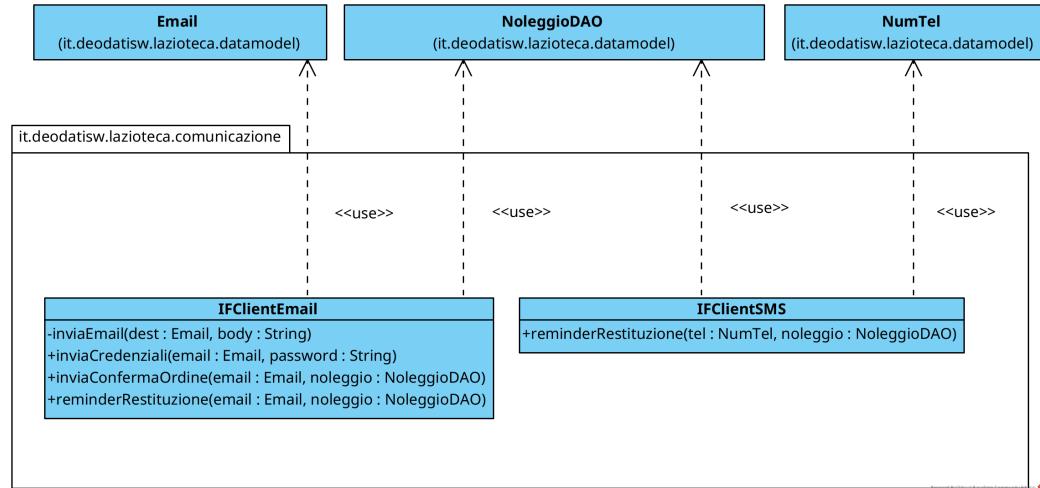
3.3 Package it.deodatisw.lazioteca.datamodel



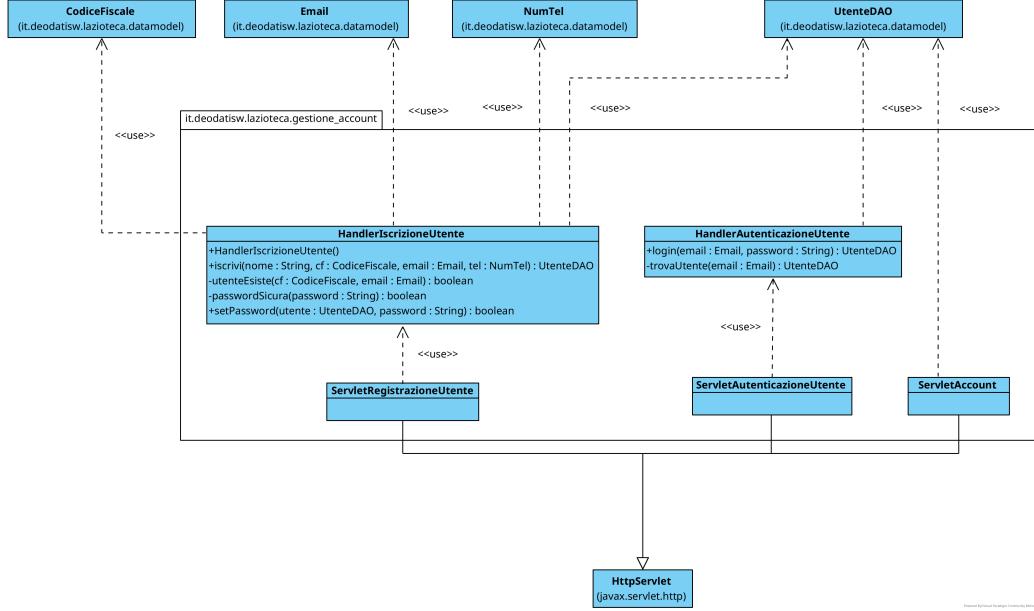
3.4 Package it.deodatisw.lazioteca.ui



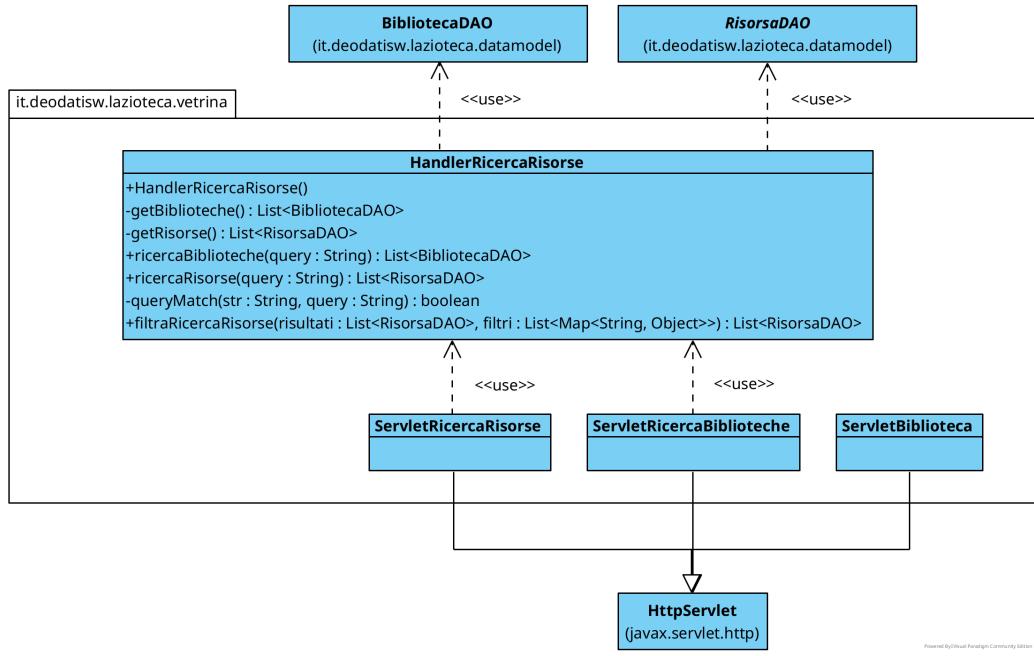
3.5 Package it.deodatisw.lazioteca.comunicazione



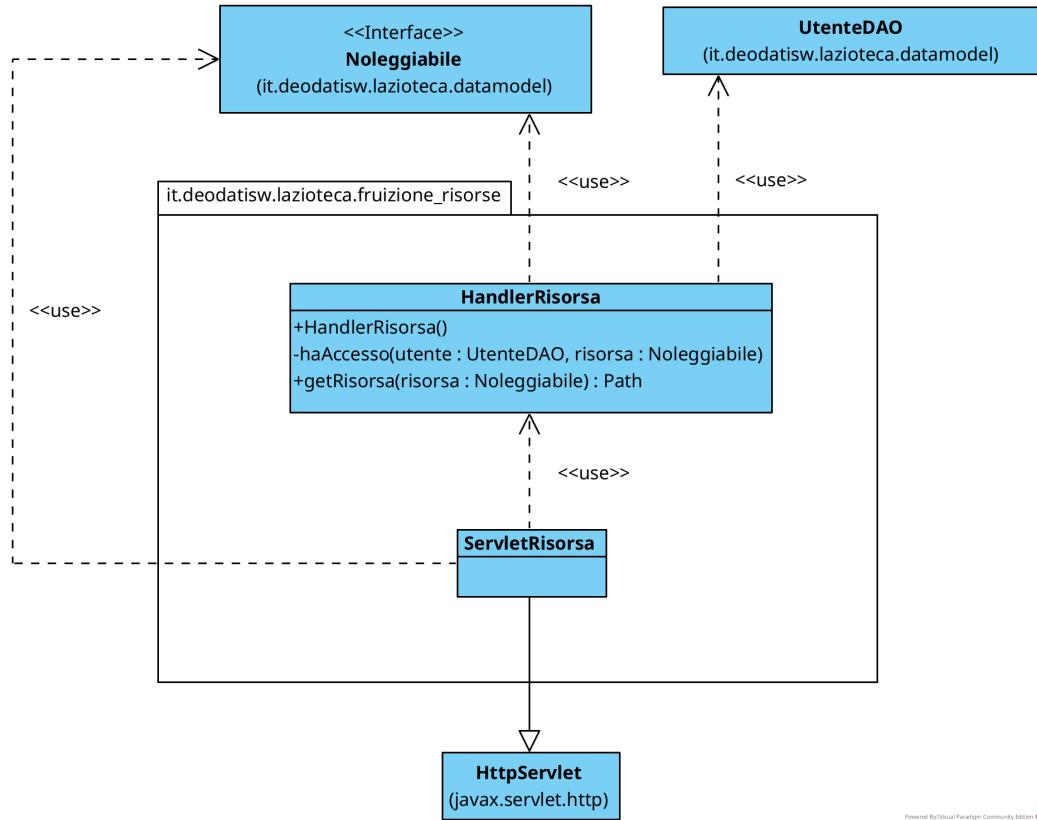
3.6 Package it.deodatisw.lazioteca.gestione_account



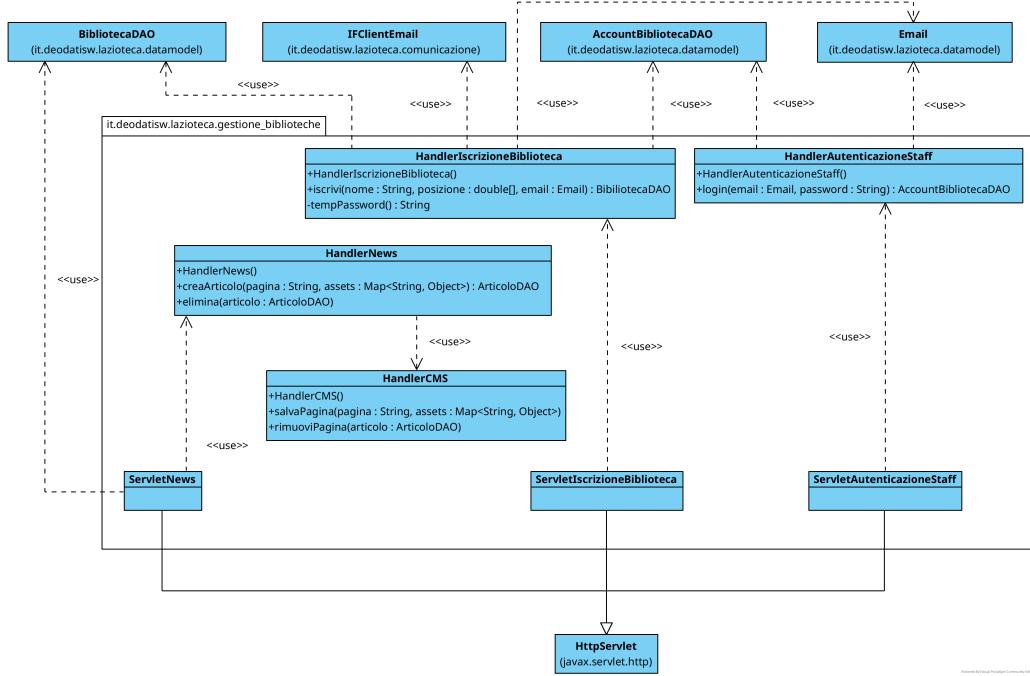
3.7 Package it.deodatisw.lazioteca.vetrina



3.8 Package it.deodatisw.lazioteca.fruizione_risorse

Powered By/Visual Paradigm Community Edition

3.9 Package it.deodatisw.lazioteca.gestione_biblioteche

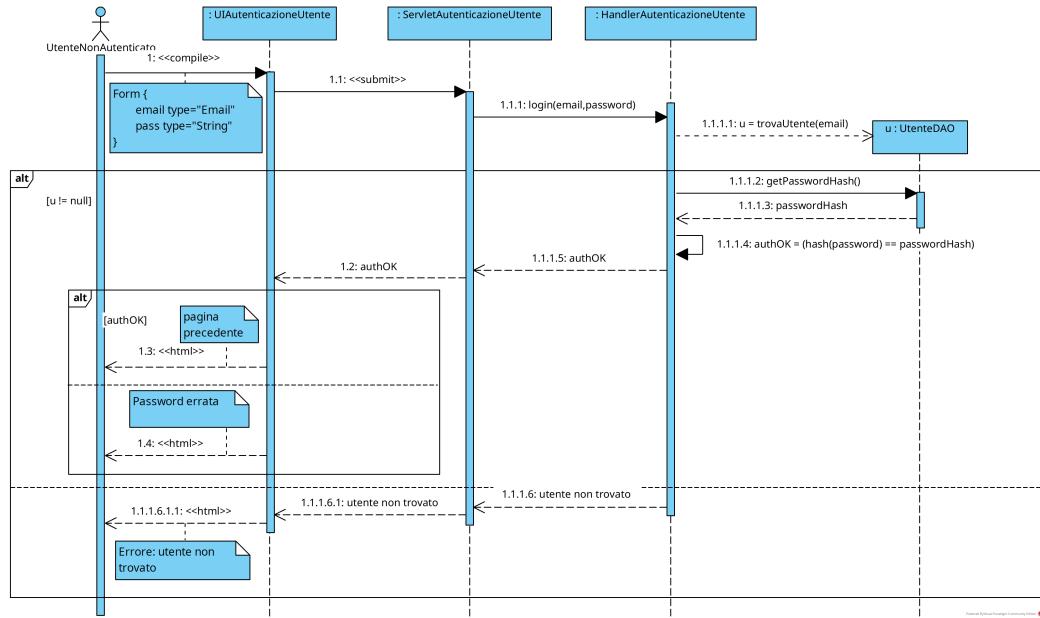


4 Diagrammi di sequenza di Design

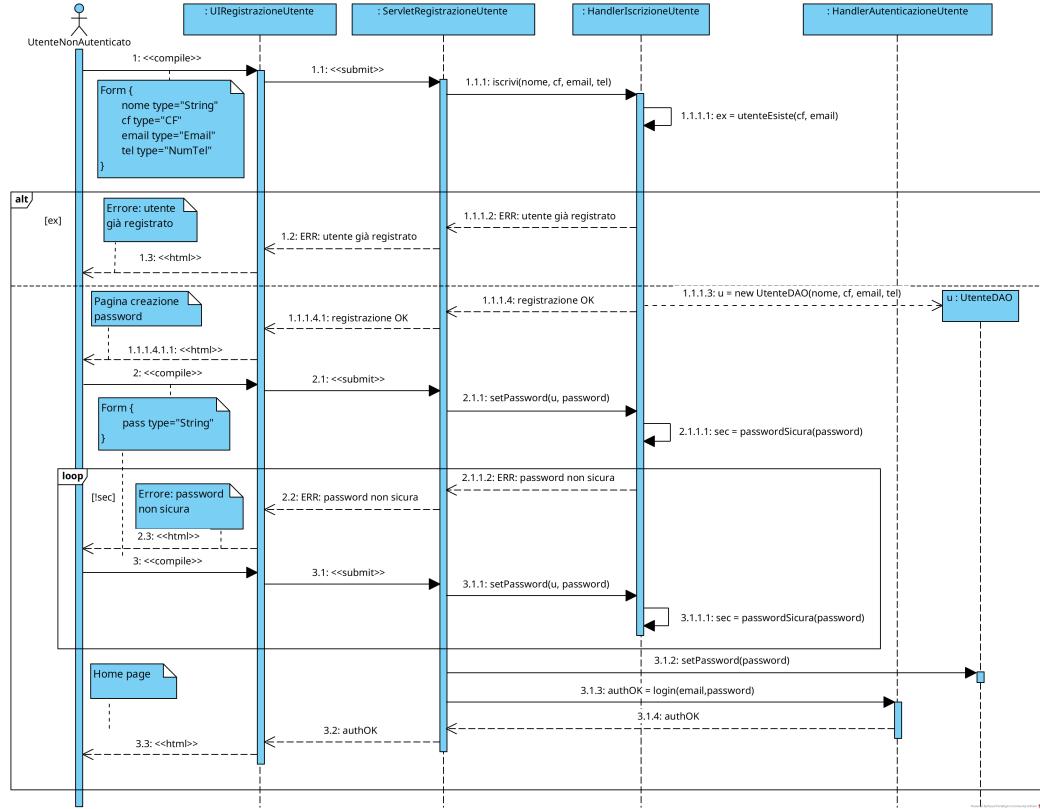
Di seguito i diagrammi di sequenza rielaborati secondo le classi di design. I diagrammi sono identificati come `SDDN`, dove N è il numero dello use-case realizzato. La logica di accesso al database e di routing dell'API sono omesse.

4.1 GestioneAccount

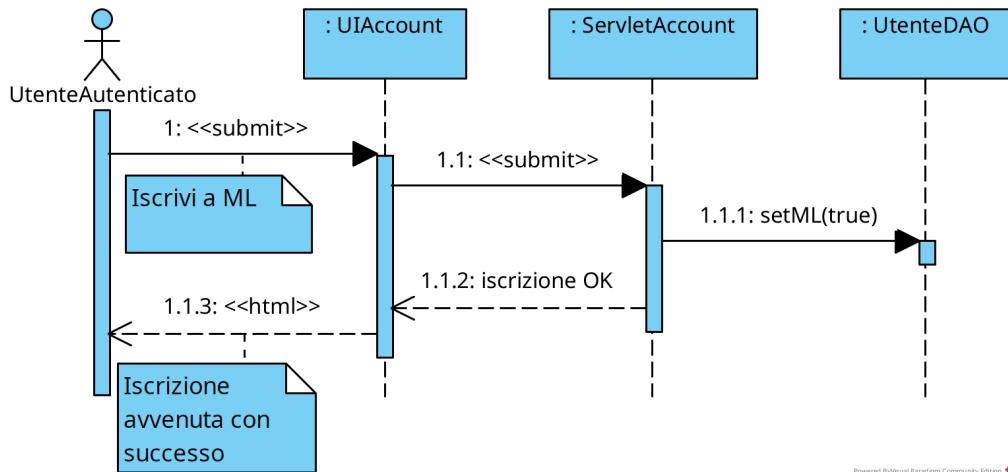
4.1.1 SDD01



4.1.2 SDD02

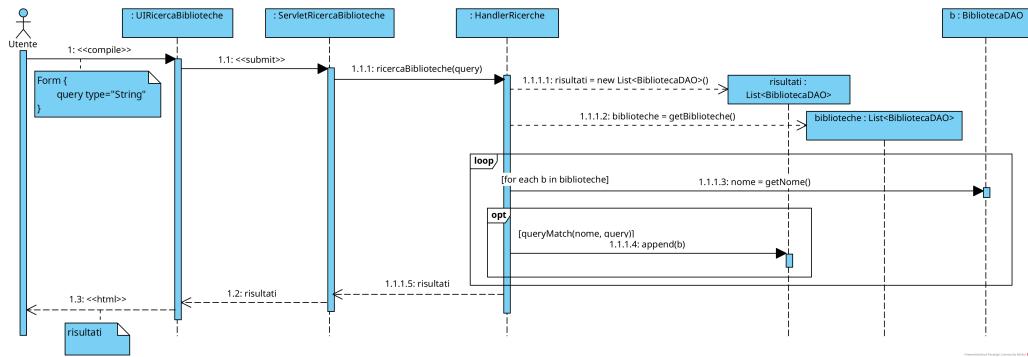


4.1.3 SDD22

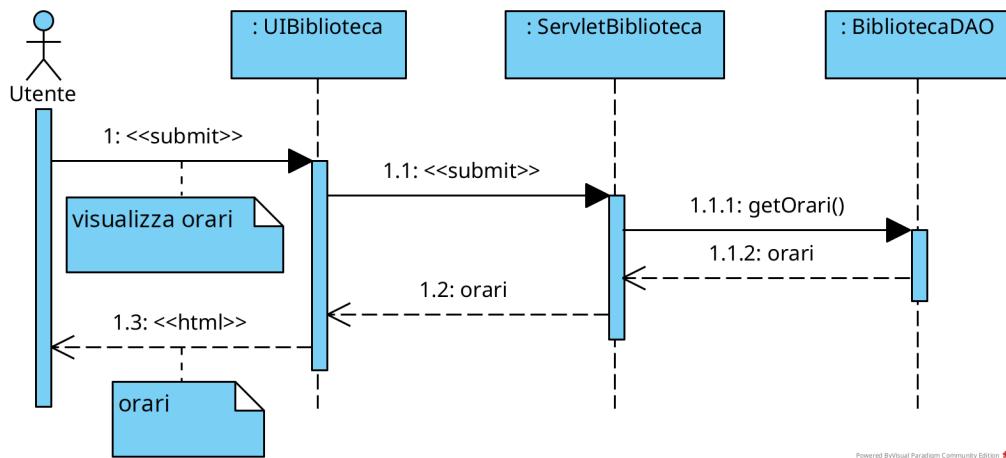


4.2 Vetrina

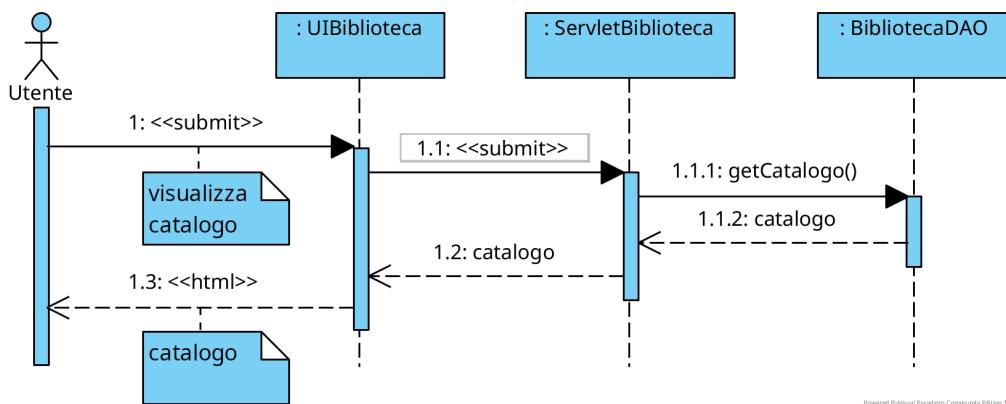
4.2.1 SDD03



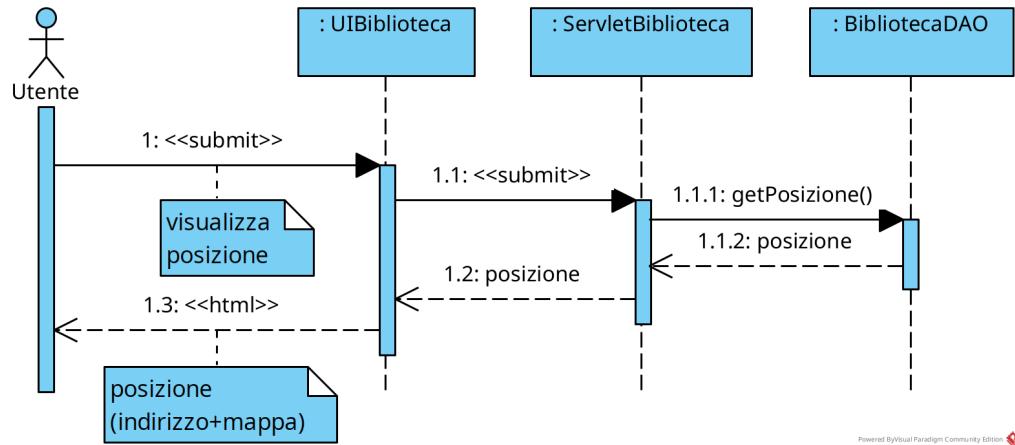
4.2.2 SDD04



4.2.3 SDD05

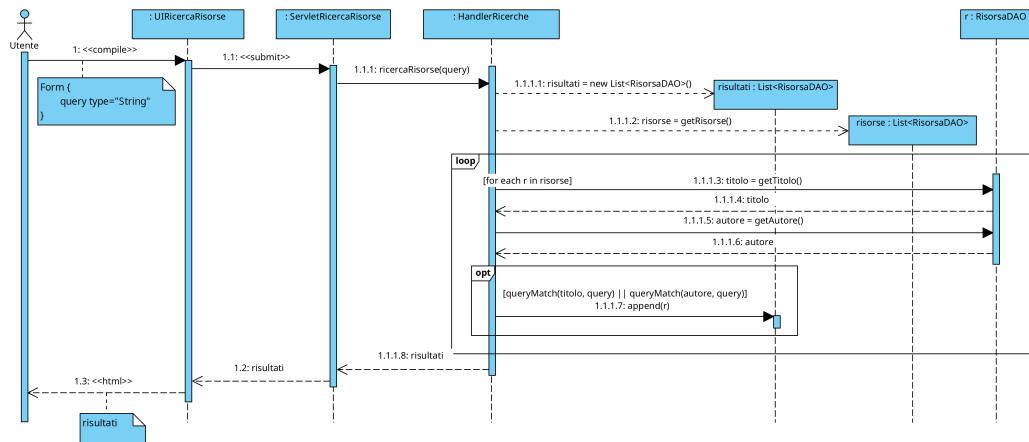


4.2.4 SDD06

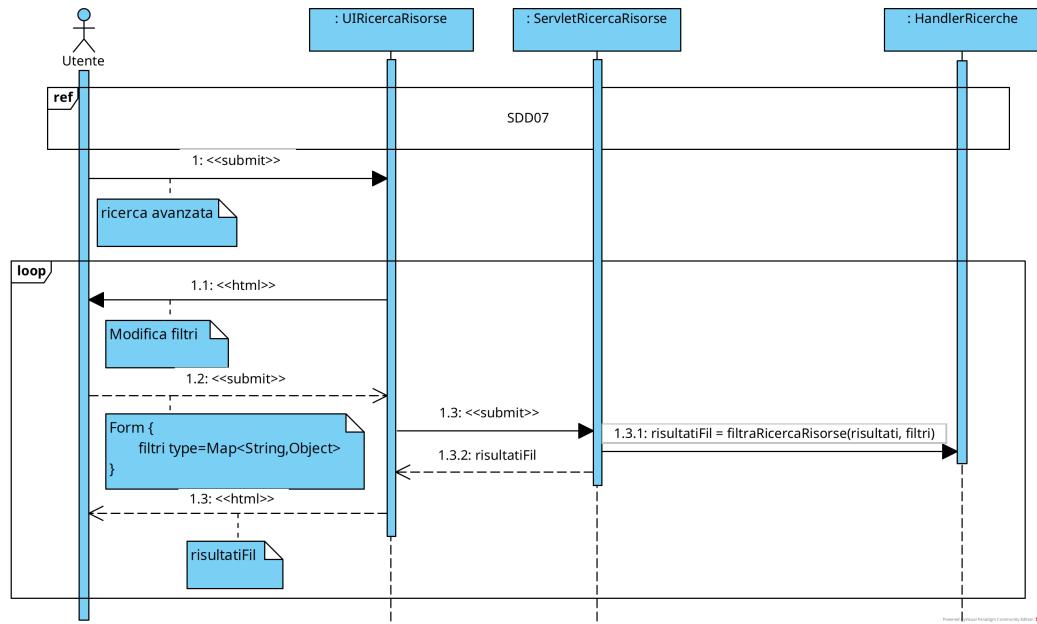


Powered ByVisual Paradigm Community Edition

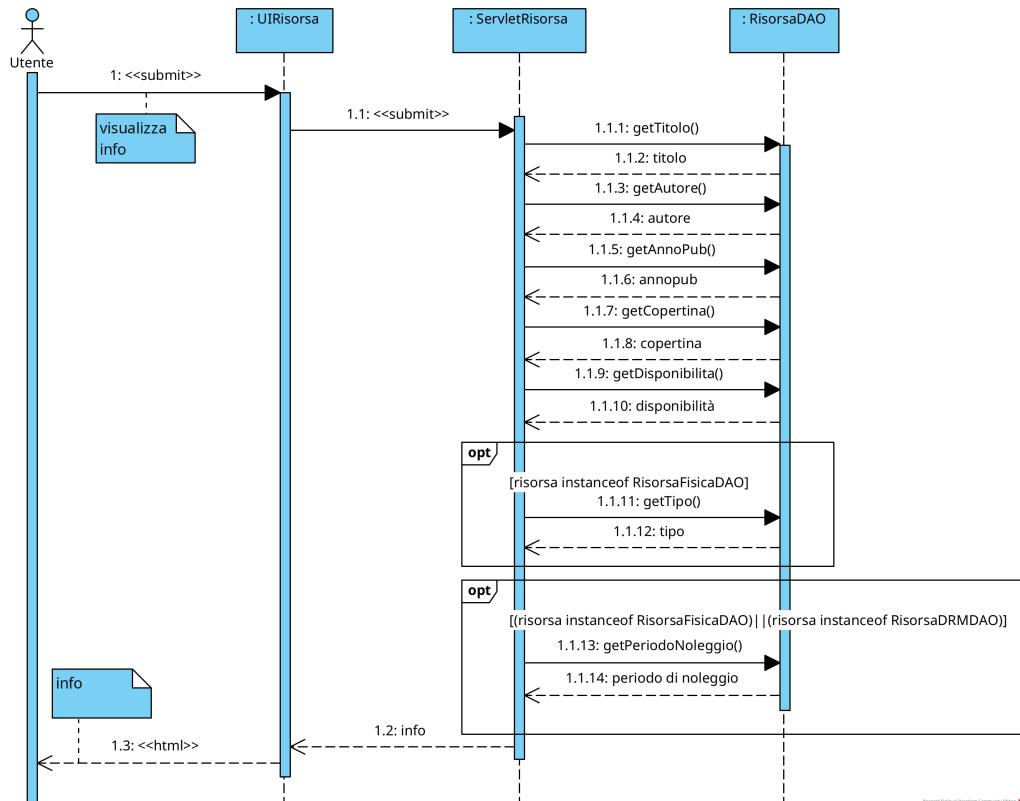
4.2.5 SDD07



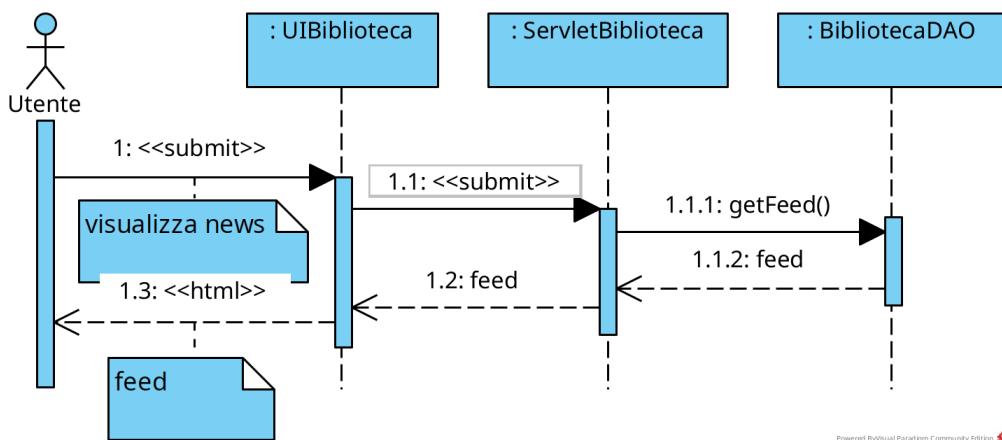
4.2.6 SDD08



4.2.7 SDD09

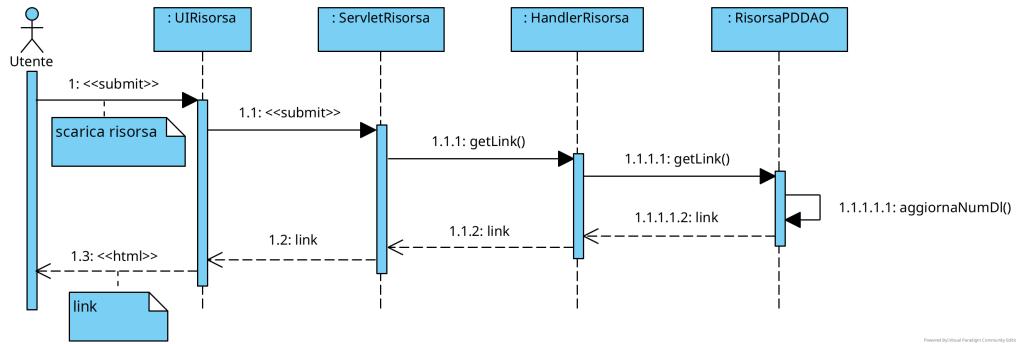


4.2.8 SDD28

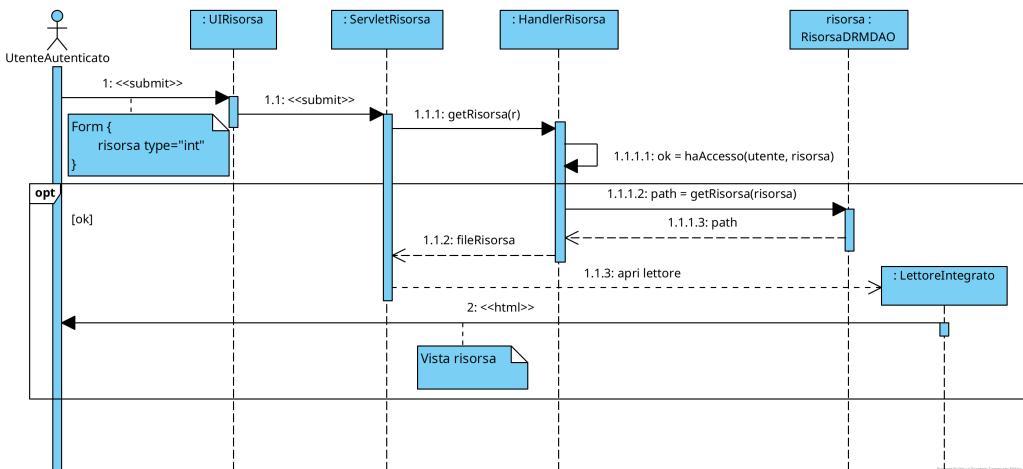


4.3 FruizioneRisorse

4.3.1 SDD10

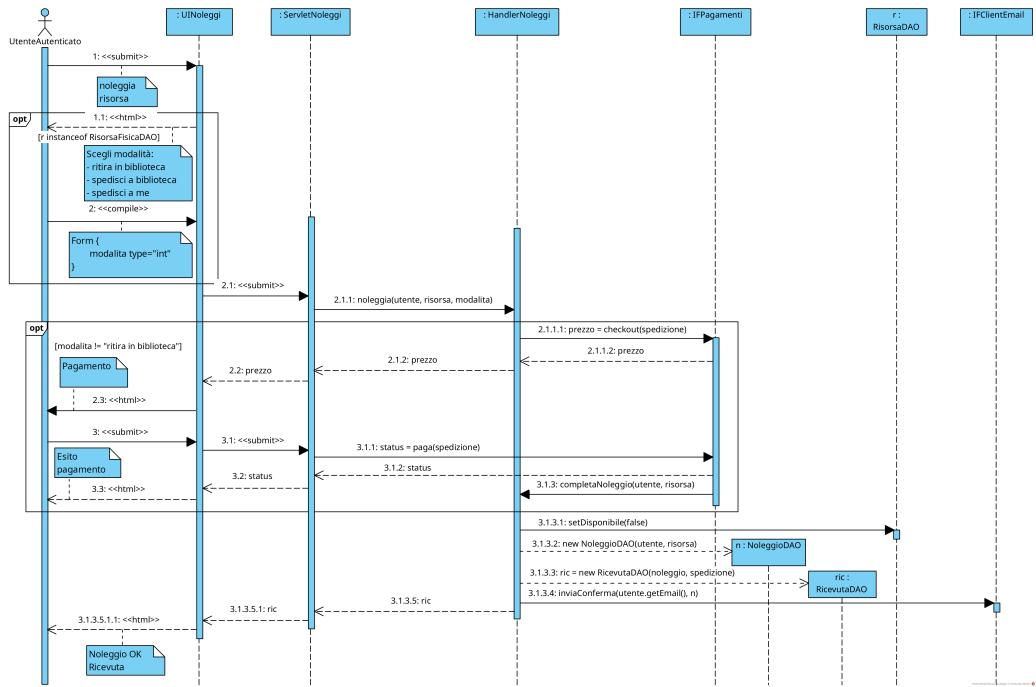


4.3.2 SDD11

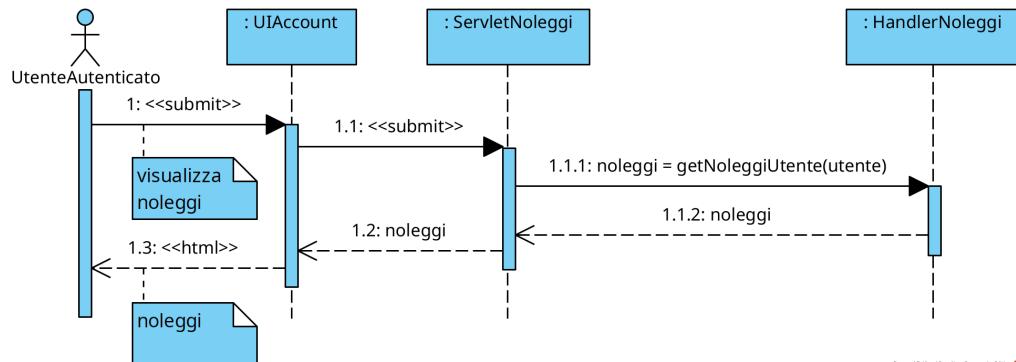


4.4 Noleggi

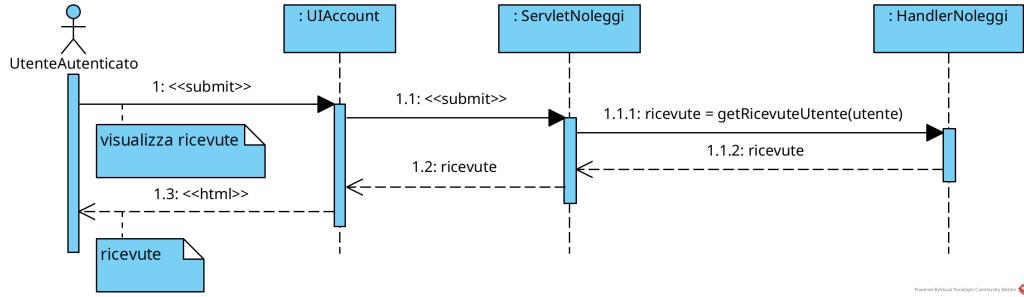
4.4.1 SDD12



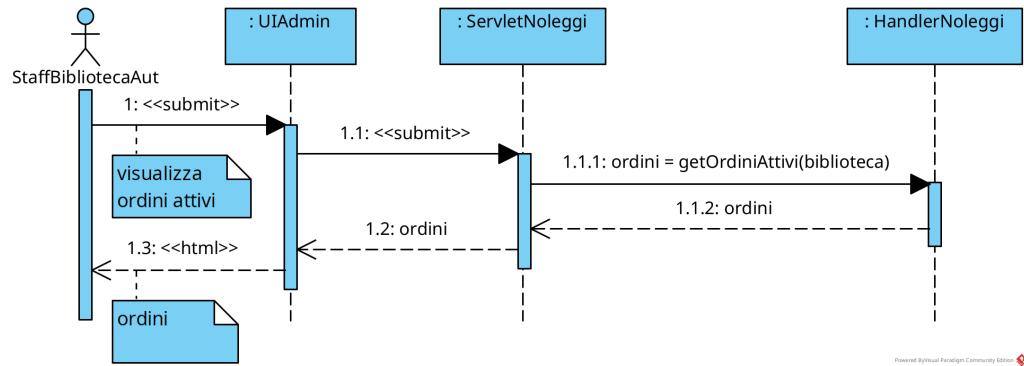
4.4.2 SDD13



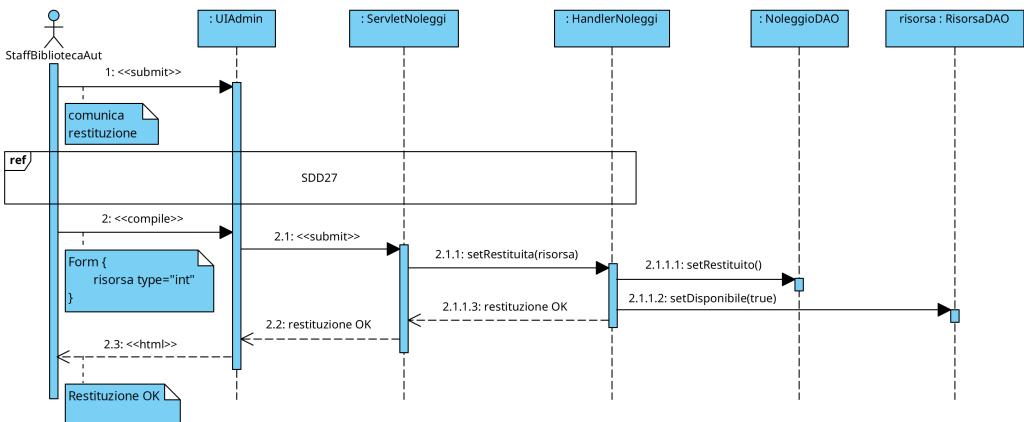
4.4.3 SDD14



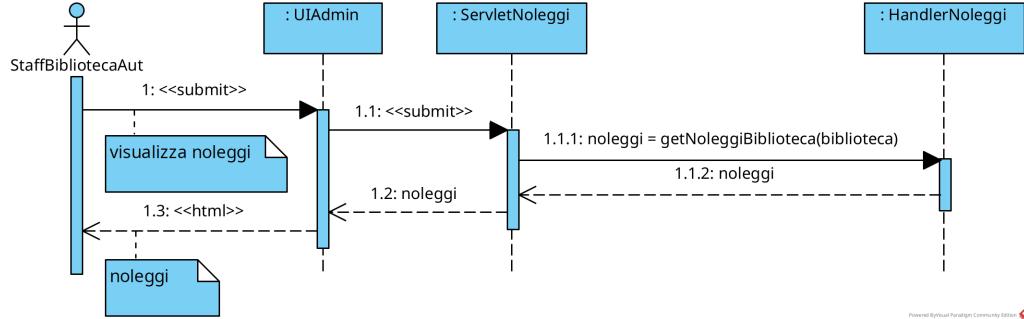
4.4.4 SDD15



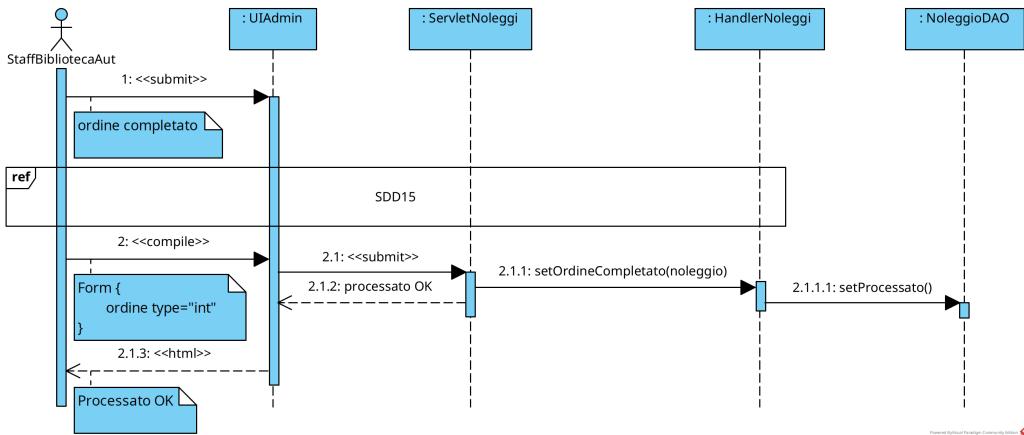
4.4.5 SDD26



4.4.6 SDD27

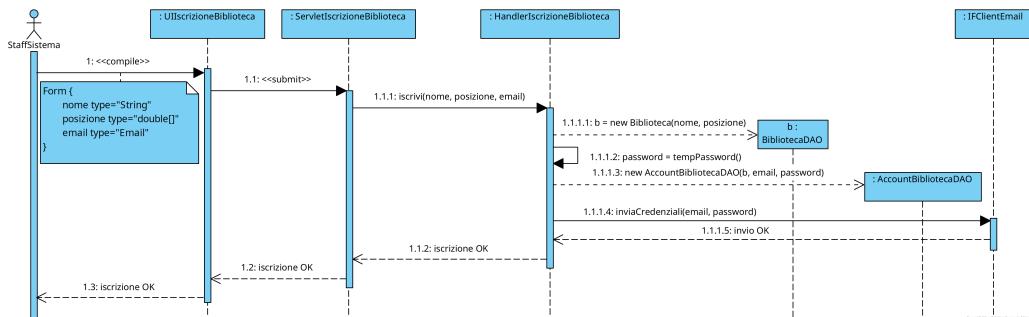


4.4.7 SDD29

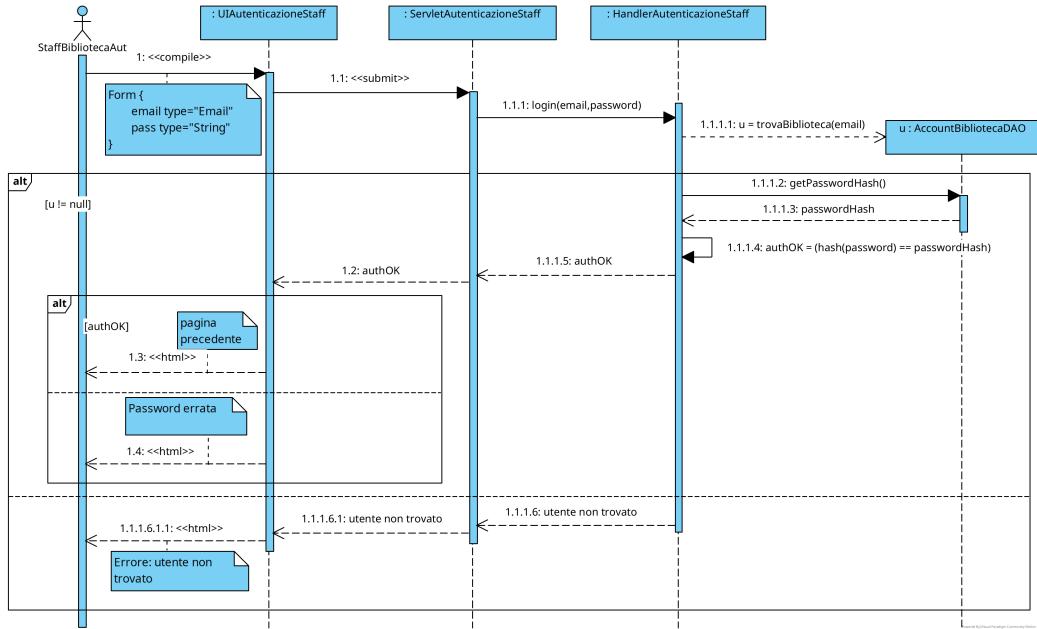


4.5 GestioneBiblioteche

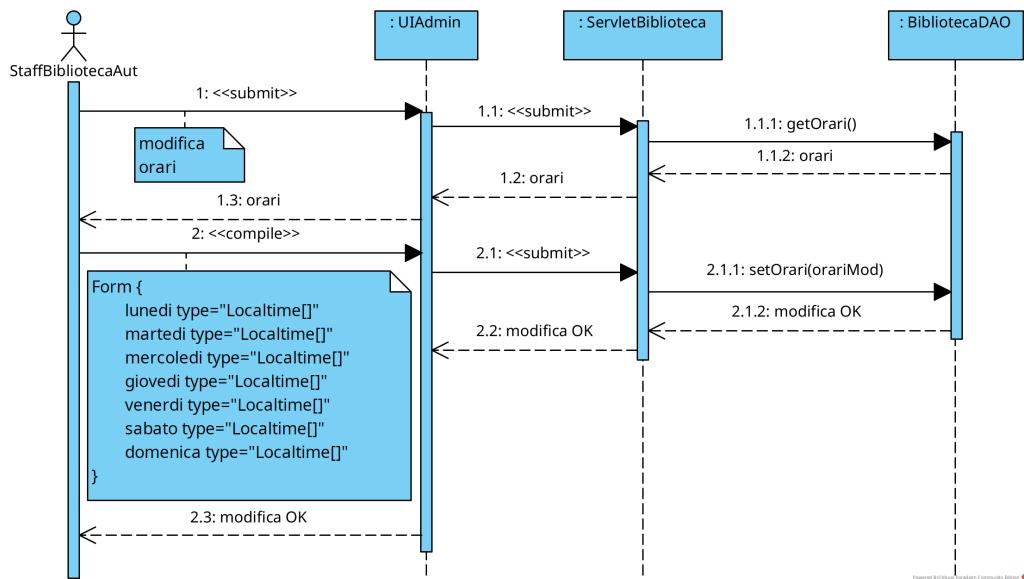
4.5.1 SDD16



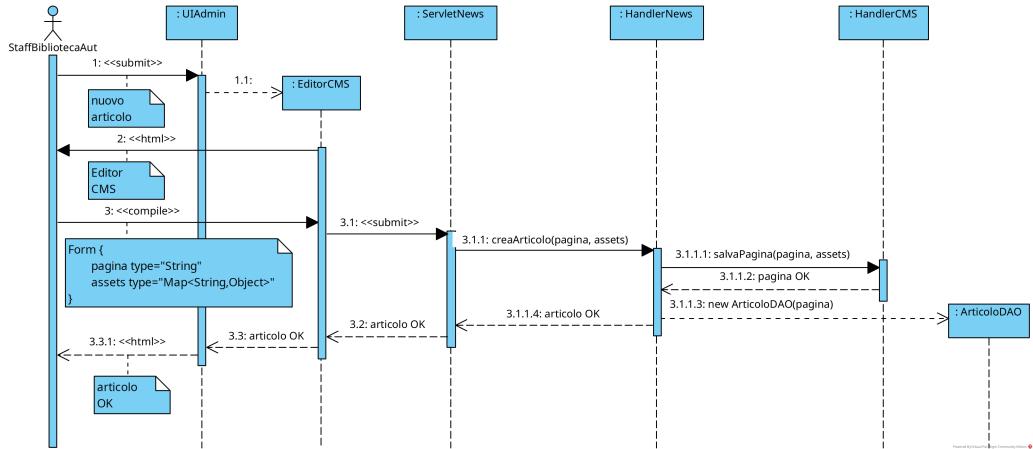
4.5.2 SDD17



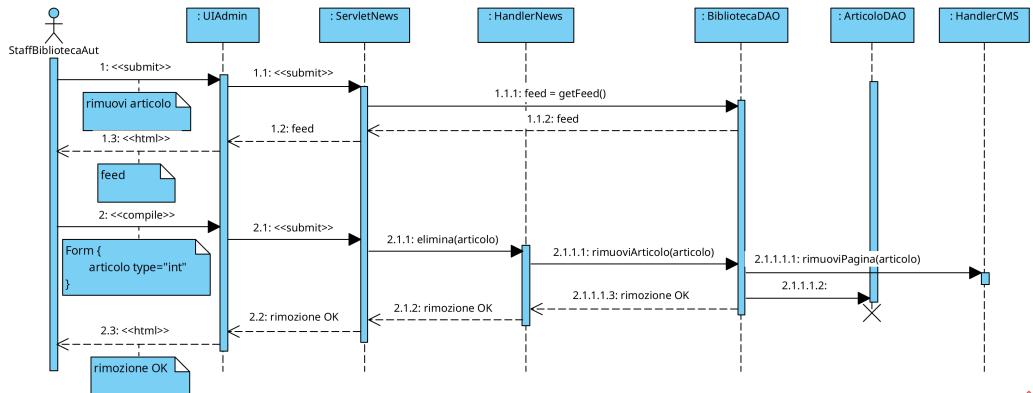
4.5.3 SDD18



4.5.4 SDD19

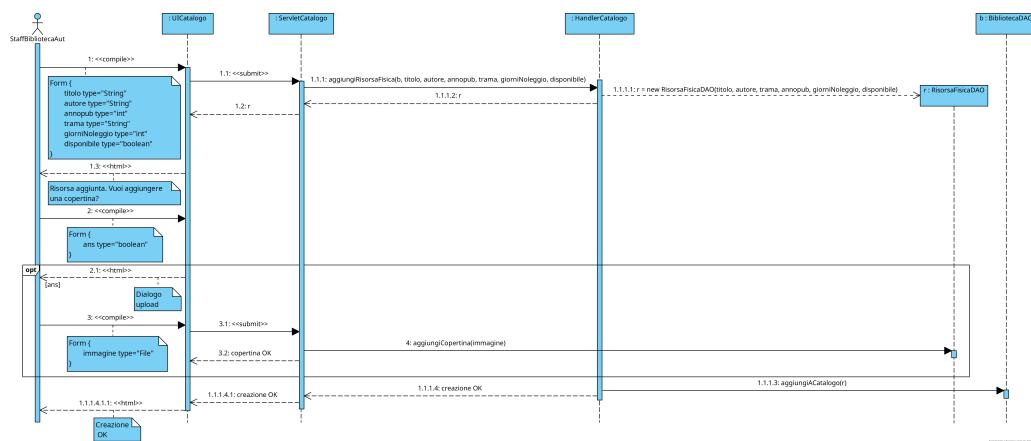


4.5.5 SDD20

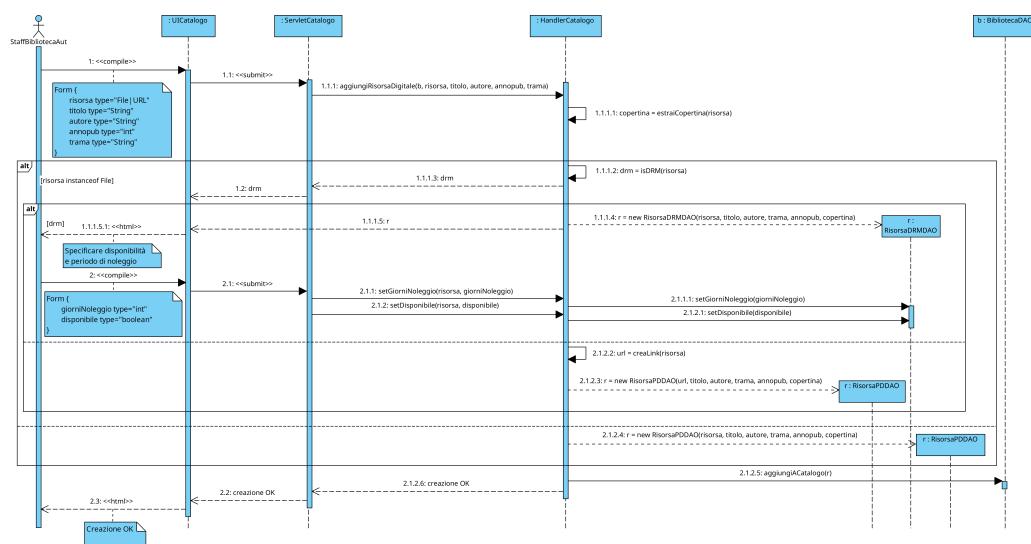


4.6 GestioneCatalogo

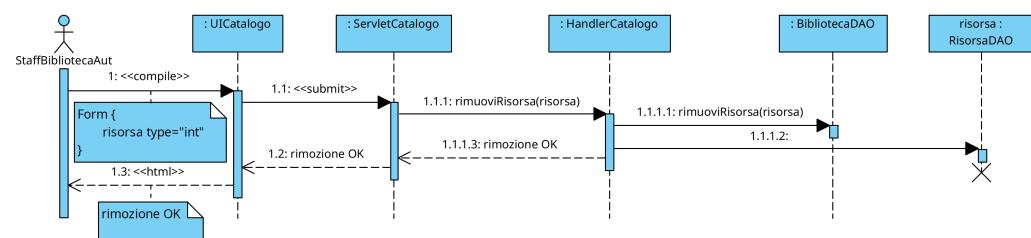
4.6.1 SDD23



4.6.2 SDD24



4.6.3 SDD25



4.6.4 SDD30

