Università degli Studi di Verona
Dipartimento di Informatica
A.A. 2018-2019

DOCUMENTAZIONE PER IL PROGETTO DI "INGEGNERIA DEL SOFTWARE"

Berardo Raffaele – VR

Zampieri Davide – VR421649

Sommario

Specifica del progetto

Si vuole progettare un sistema informatico per gestire gli acquisti on-line di una libreria.

Per ogni libro si memorizza il titolo, l'autore o gli autori, la casa editrice, l'anno di pubblicazione, il codice ISBN (identificativo), il genere, il prezzo ed una breve descrizione. Gli utenti possono visualizzare le classifiche di vendita che sono organizzate per genere (novità, narrativa, ragazzi, …) e vengono aggiornate ogni settimana. Per ogni posizione della classifica si indica da quante settimane il libro è in quella posizione.

Il sistema memorizza gli ordini degli utenti. Gli utenti possono essere registrati o meno. Per gli utenti si memorizzano nome, cognome, indirizzo, CAP, città, numero di telefono ed e-mail. Ogni utente registrato accede con e-mail e password ed ha associata una LibroCard per la raccolta punti. Ogni LibroCard ha un numero identificativo, una data di emissione e il totale dei punti raccolti. Gli utenti registrati possono specificare uno o più indirizzi di spedizione diversi da quello di residenza.

Ogni libro ha associato il numero di punti che vengono caricati sulle LibroCard in caso di acquisto da parte di utenti registrati.

Per ogni ordine si memorizzano il codice (univoco), la data, i libri che lo compongono, l'utente che lo ha effettuato, il costo totale, il tipo di pagamento (carta di credito, PayPal o contrassegno) e il saldo punti se l'utente è registrato.

Il sistema deve permettere agli utenti registrati di accedere al loro profilo, modificare i dati anagrafici, verificare il saldo punti e lo stato dei loro ordini. Ogni utente registrato può vedere tutti gli ordini che ha effettuato nel tempo con il totale dei punti accumulati per ogni ordine. Gli utenti non registrati possono accedere agli ordini che hanno effettuato tramite il codice dell'ordine.

I responsabili della libreria devono poter verificare lo stato degli ordini e il saldo punti delle LibroCard degli utenti registrati. Inoltre, i responsabili della libreria sono responsabili dell'inserimento dei dati relativi ai libri che si possono ordinare e dell'aggiornamento delle classifiche.

Tutti gli utenti sono opportunamente autenticati dal sistema per poter accedere alle funzionalità di loro competenza.

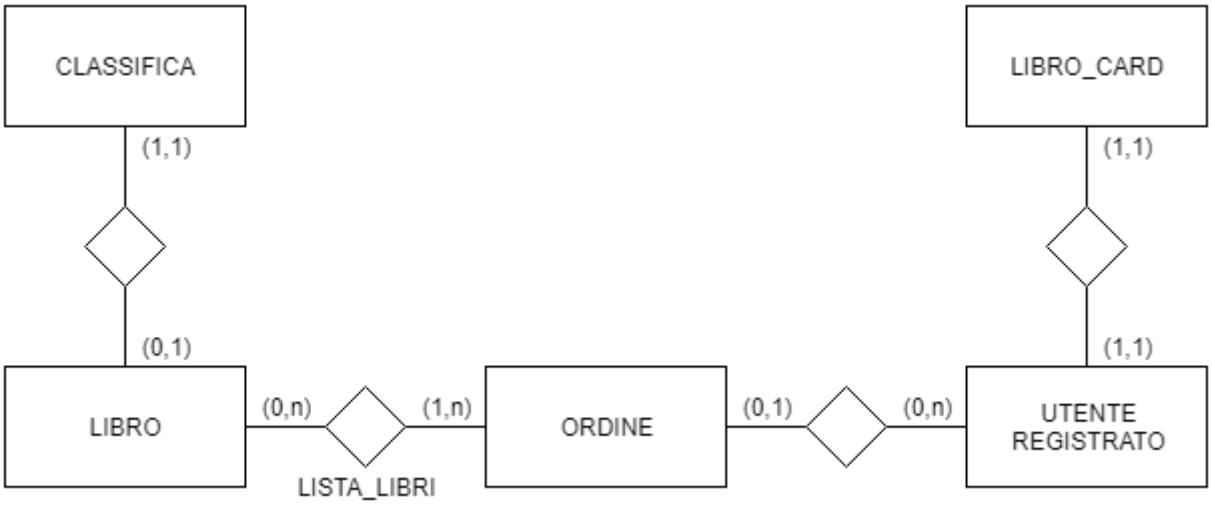
Introduzione

Software utilizzati

Il progetto è stato sviluppato utilizzando l'IDE di *Eclipse* con la versione di Java 8. Per lo sviluppo dell'interfaccia grafica si è fatto uso delle librerie del framework *Swing*. Inoltre, la documentazione del codice sorgente è stata scritta utilizzando *JavaDoc*. Infine, per disegnare i diagrammi UML è stato usato il software *StarUML*.

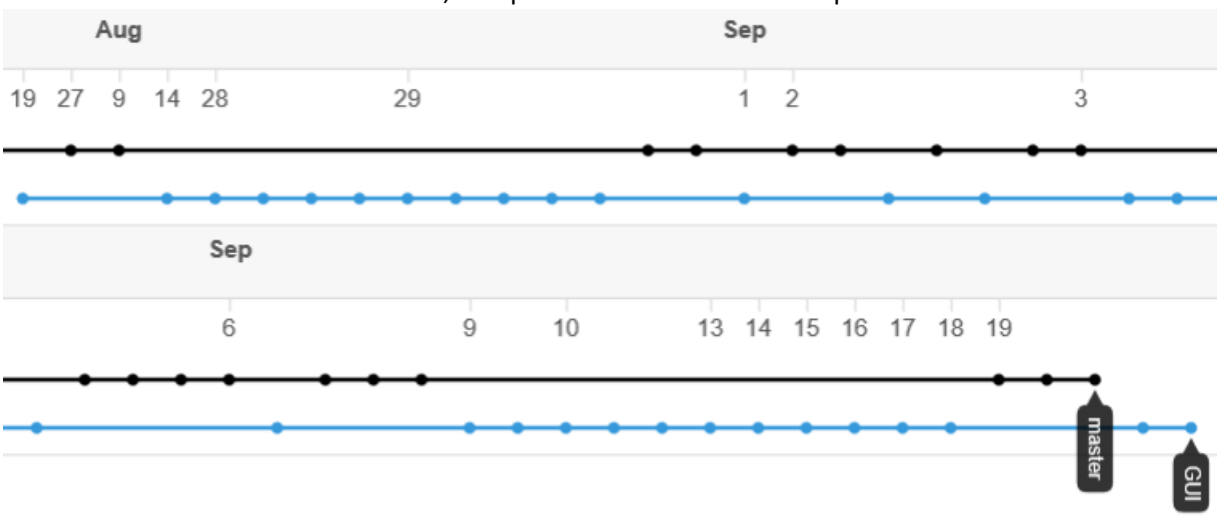
Gestione dei dati

Per salvare e gestire i dati necessari al funzionamento di questo progetto è stato scelto di incorporare un database *Apache Derby*, il quale ha permesso l'accesso ai dati tramite il linguaggio universale SQL e una maggiore efficienza dovuta anche al controllo della ridondanza. Di seguito viene mostrata la *struttura* data al database dopo aver individuato tutte le entità presenti nella specifica:



Controllo della versione

Per il controllo della versione del codice è stato usato il software *GIT* (tramite GitHub). Tale strumento ha permesso la creazione di diverse *branch*, utili per la suddivisione dei compiti.



Diagrammi dei casi d'uso

Note

I *diagrammi dei casi d'uso* individuano chi ha a che fare con il sistema (attore) e che cosa l'attore può fare (caso d'uso). I *requisiti* presentati nella specifica sono tutti *funzionali* in quanto specificano cosa deve essere fatto. I *casi d'uso* modellano i requisiti funzionali perché specificano cosa ci si aspetta da un sistema ma nascondono come il sistema lo implementa.

Attori

Dall'analisi della specifica si individuano tre tipologie di attori, ognuna con il proprio insieme di azioni:

- *Utenti registrati*, che possono visualizzare il loro profilo, modificare i dati anagrafici, verificare il saldo punti, verificare lo stato dei loro ordini visualizzando tutti gli ordini effettuati nel tempo con il totale dei punti accumulati per ogni ordine.
- *Utenti non registrati*, che possono accedere agli ordini effettuati.
- *Responsabili della libreria*, che possono verificare lo stato degli ordini di tutti gli utenti, verificare il saldo punti delle LibroCard degli utenti registrati, inserire i dati relativi ai libri e aggiornare le classifiche.

Inoltre, tutti gli attori sono opportunamente autenticati dal sistema per poter accedere alle azioni di loro competenza.

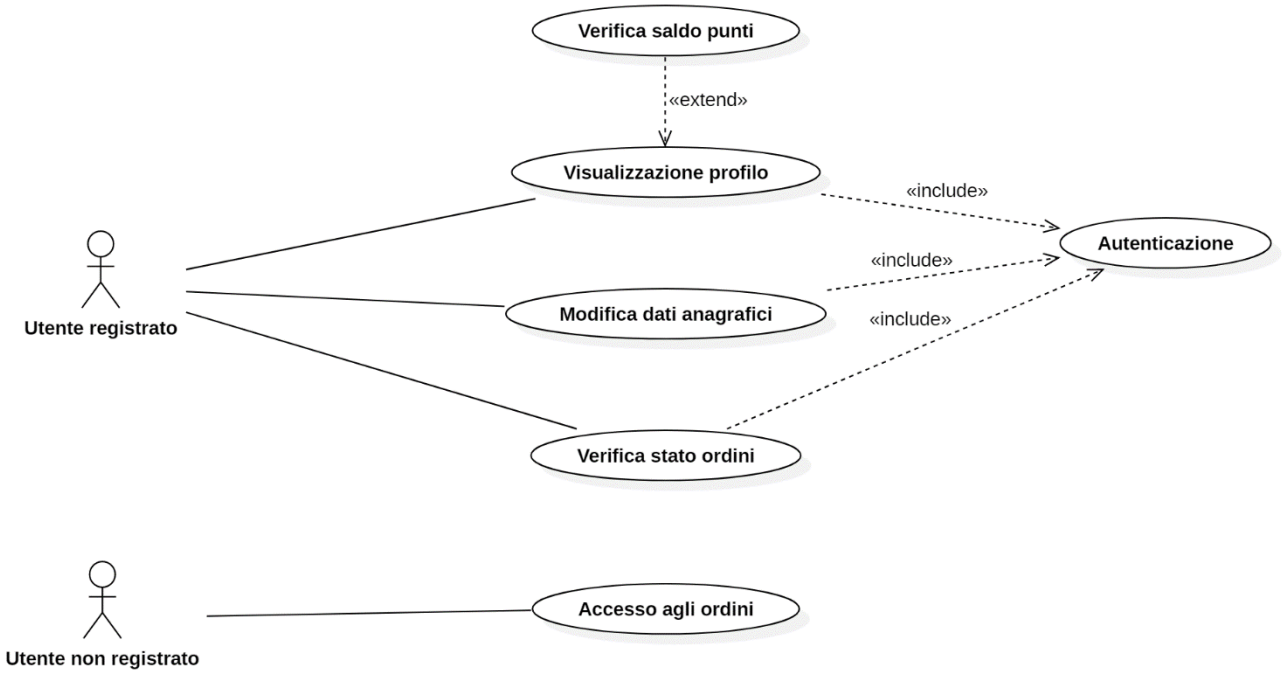
Dati già presenti

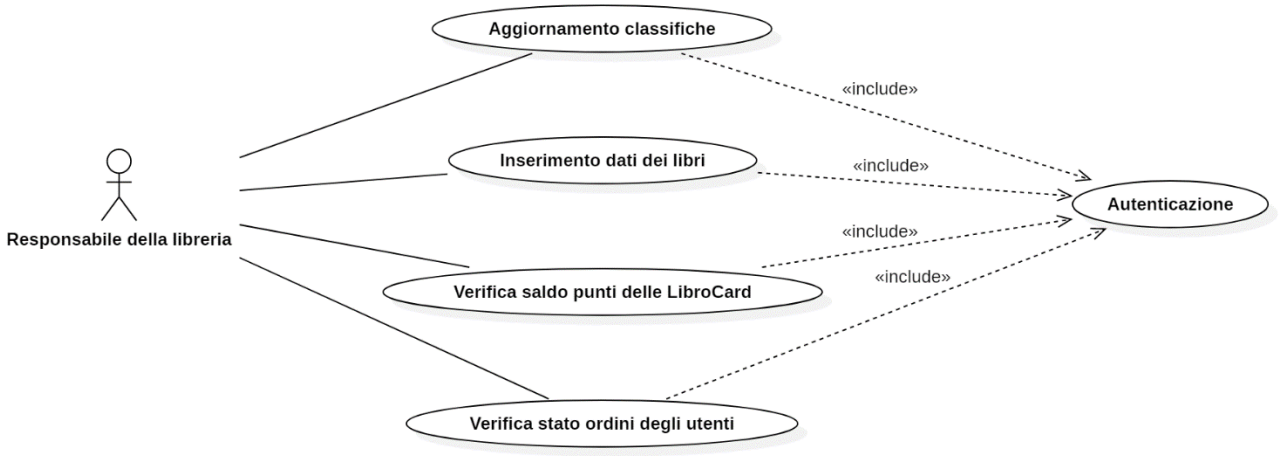
Per provare le funzionalità relative ai diversi tipi di attori sono stati introdotti nel sistema i seguenti dati:

- Un utente registrato, con email "*zampierida98@gmail.com*" e password "*univr*".
- Un responsabile della libreria, con username "*admin*" e password "*admin*".

Inoltre, l'utente registrato di cui sopra ha una LibroCard associata ed ha già effettuato un ordine.

Diagrammi





Schede di specifica

Caso d'uso: Visualizzazione profilo
ID: UC-1
Attori: Utente registrato
Precondizioni: Autenticazione
Sequenza degli eventi: 1. Click sul bottone "Visualizza profilo"; 2. Recupero dati dal database; 3. Restituzione dei dati raccolti all'utente.
Postcondizioni: Nessuna

Caso d'uso: Modifica dati anagrafici
ID: UC-2
Attori: Utente registrato
Precondizioni: Autenticazione
Sequenza degli eventi: 1. Click sul bottone "Modifica profilo"; 2. Visualizzazione dell'interfaccia per la modifica del profilo; 3. Inserimento dei dati nei campi che si vogliono modificare; 4. Click sul bottone "Modifica Dati"; 5. Controllo dei dati inseriti: a. Se sono corretti, invio i dati al database e restituisco un messaggio di modifiche effettuate; b. Se non sono corretti, restituisco un messaggio di modifiche non effettuate.
Postcondizioni: I dati anagrafici dell'utente sono stati aggiornati

Caso d'uso: Verifica saldo punti
ID: UC-3
Attori: Utente registrato
Precondizioni: Autenticazione
Sequenza degli eventi: 1. Click sul bottone "Visualizza profilo"; 2. Recupero dati dal database; 3. Visualizzazione dei dati relativi al profilo e alla LibroCard.
Postcondizioni: Nessuna

Caso d'uso: Verifica stato ordini
ID: UC-4
Attori: Utente registrato
Precondizioni: Autenticazione
Sequenza degli eventi: 1. Click sul bottone "Visualizza ordini"; 2. Recupero dati degli ordini dal database; 3. Visualizzazione della tabella di tutti gli ordini effettuati nel tempo con il totale dei punti accumulati.
Postcondizioni: Nessuna

Caso d'uso: Accesso agli ordini
ID: UC-5
Attori: Utente non registrato
Precondizioni: Possedere il codice di un ordine effettuato
Sequenza degli eventi: 1. Click sul bottone "Non registrati"; 2. Inserimento del codice dell'ordine nell'apposito campo; 3. Click sul bottone "Accedi all'ordine"; 4. Controllo sulla validità del codice: a. Se è valido, recupero i dati dal database e visualizzo la tabella con i dati dell'ordine; b. Se non è valido, cancello il codice inserito.
Postcondizioni: Nessuna

Caso d'uso: Verifica stato ordini degli utenti
ID: UC-6
Attori: Responsabile della libreria
Precondizioni: Autenticazione in area riservata
Sequenza degli eventi: 1. Click sul bottone "Ordini"; 2. Seleziono tramite il menu a tendina l'utente di cui voglio visualizzare gli ordini; 3. Recupero i dati degli ordini di quell'utente dal database; 4. Visualizzo la tabella contenente tutti gli ordini effettuati dall'utente selezionato.
Postcondizioni: Nessuna

Caso d'uso: Verifica saldo punti delle LibroCard
ID: UC-7
Attori: Responsabile della libreria
Precondizioni: Autenticazione in area riservata
Sequenza degli eventi: 1. Click sul bottone "Libro Card"; 2. Recupero dei dati di tutte le LibroCard dal database; 3. Visualizzazione della tabella contenente tutte le LibroCard degli utenti registrati.
Postcondizioni: Nessuna

Caso d'uso: Inserimento dati dei libri
ID: UC-8
Attori: Responsabile della libreria
Precondizioni: Autenticazione in area riservata
Sequenza degli eventi: 1. Click sul bottone "Aggiungi libro"; 2. Inserimento dei dati del libro negli appositi campi; 3. Click sul bottone "Inserisci libro"; 4. Controllo dei dati inseriti: a. Se sono corretti, invio i dati al database e restituisco un messaggio di inserimento effettuato; b. Se non sono corretti, restituisco un messaggio di inserimento non effettuato;
Postcondizioni: Un nuovo libro è stato aggiunto al sistema

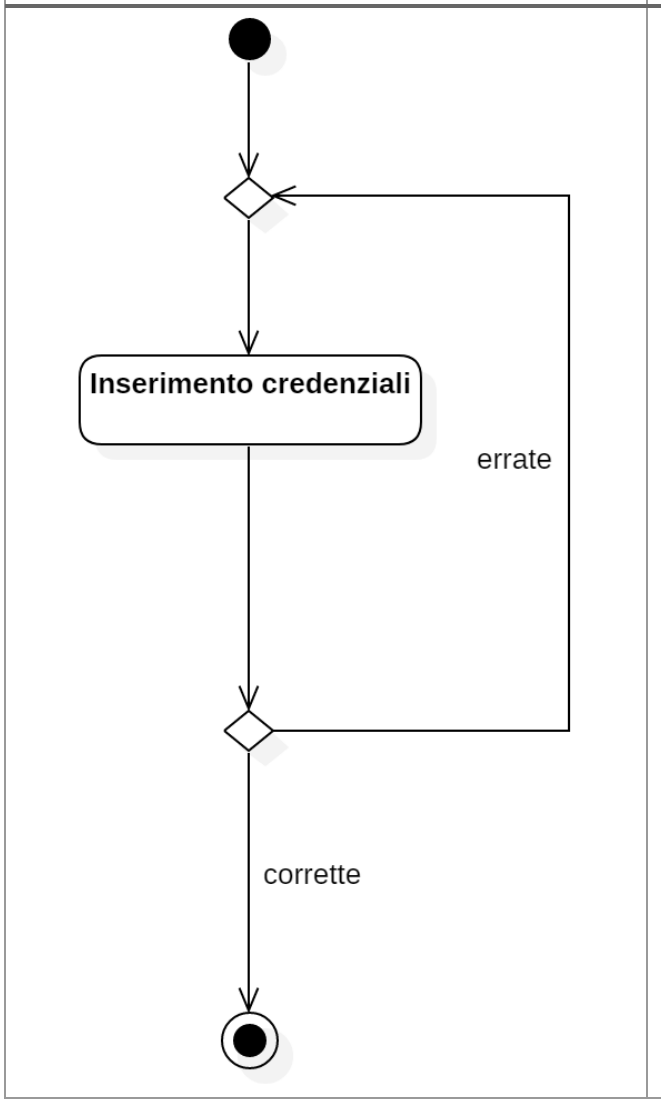
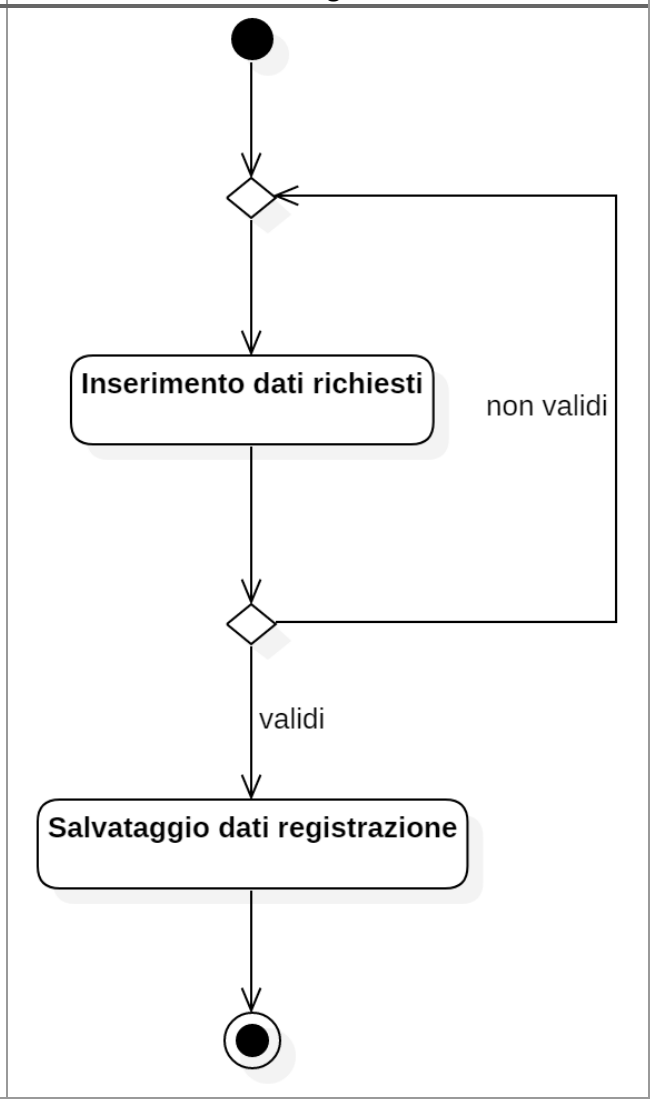
Caso d'uso: Aggiornamento classifiche
ID: UC-9
Attori: Responsabile della libreria
Precondizioni: Autenticazione in area riservata
Sequenza degli eventi: 1. Click sul bottone "Aggiorna classifiche"; 2. Seleziono tramite il menu a tendina il libro di cui voglio aggiornare la posizione in classifica; 3. Inserisco la nuova posizione in classifica nell'apposito campo; 4. Click sul bottone "Aggiorna": a. Se il dato è corretto, invio la nuova posizione al database; b. Se il dato è errato, pulisco il campo di testo.
Postcondizioni: Il libro selezionato ha una nuova posizione in classifica

Diagrammi delle attività

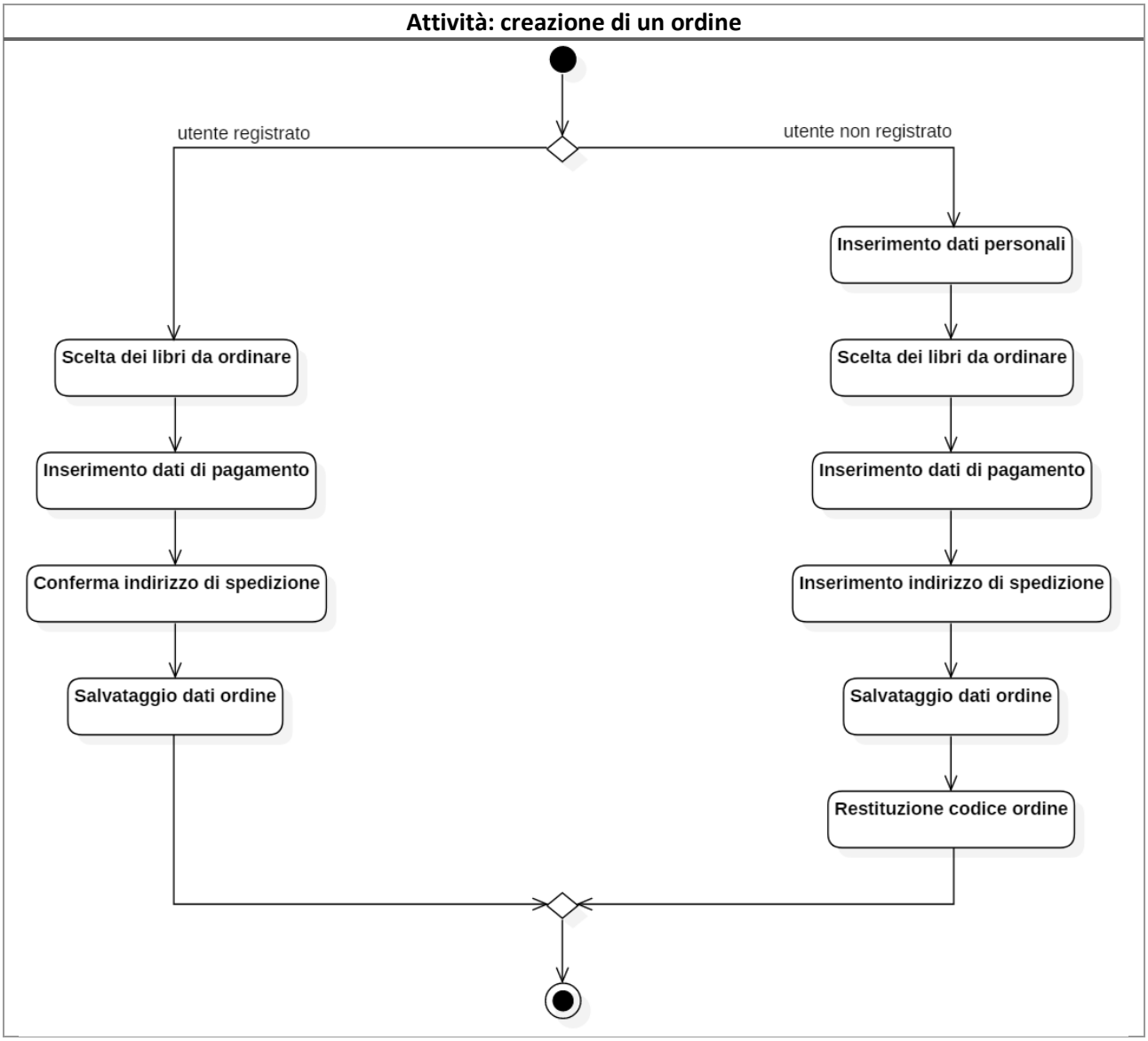
Note

I *diagrammi delle attività* modellano un comportamento (che riguarda una o più entità) come un insieme di azioni organizzate secondo un flusso. L'attività può essere relativa ad un qualsiasi *oggetto* e quindi si possono utilizzare i diagrammi delle attività per: modellare il flusso di un caso d'uso, modellare il funzionamento di un'operazione di classe o di un componente.

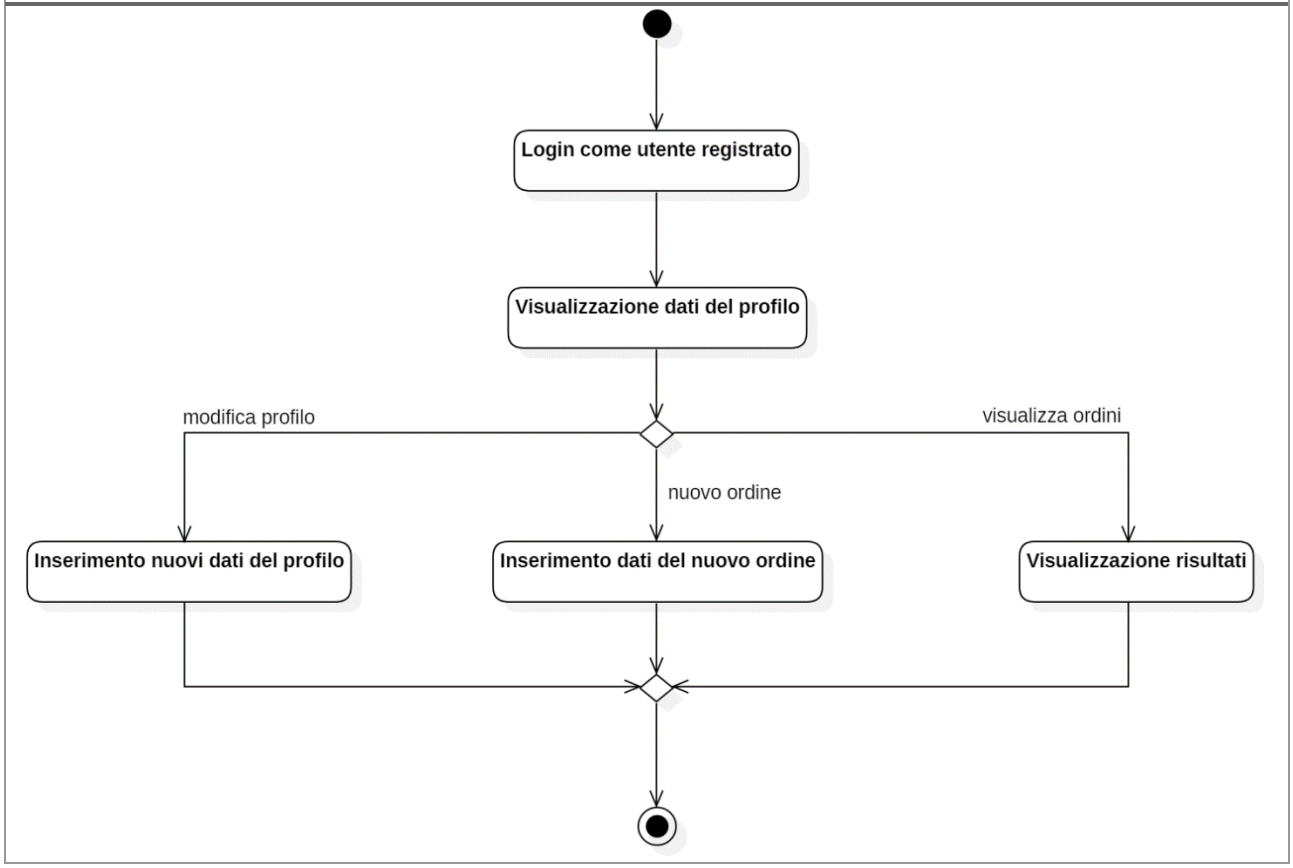
Diagrammi

Attività: autenticazione	**Attività: registrazione**
	

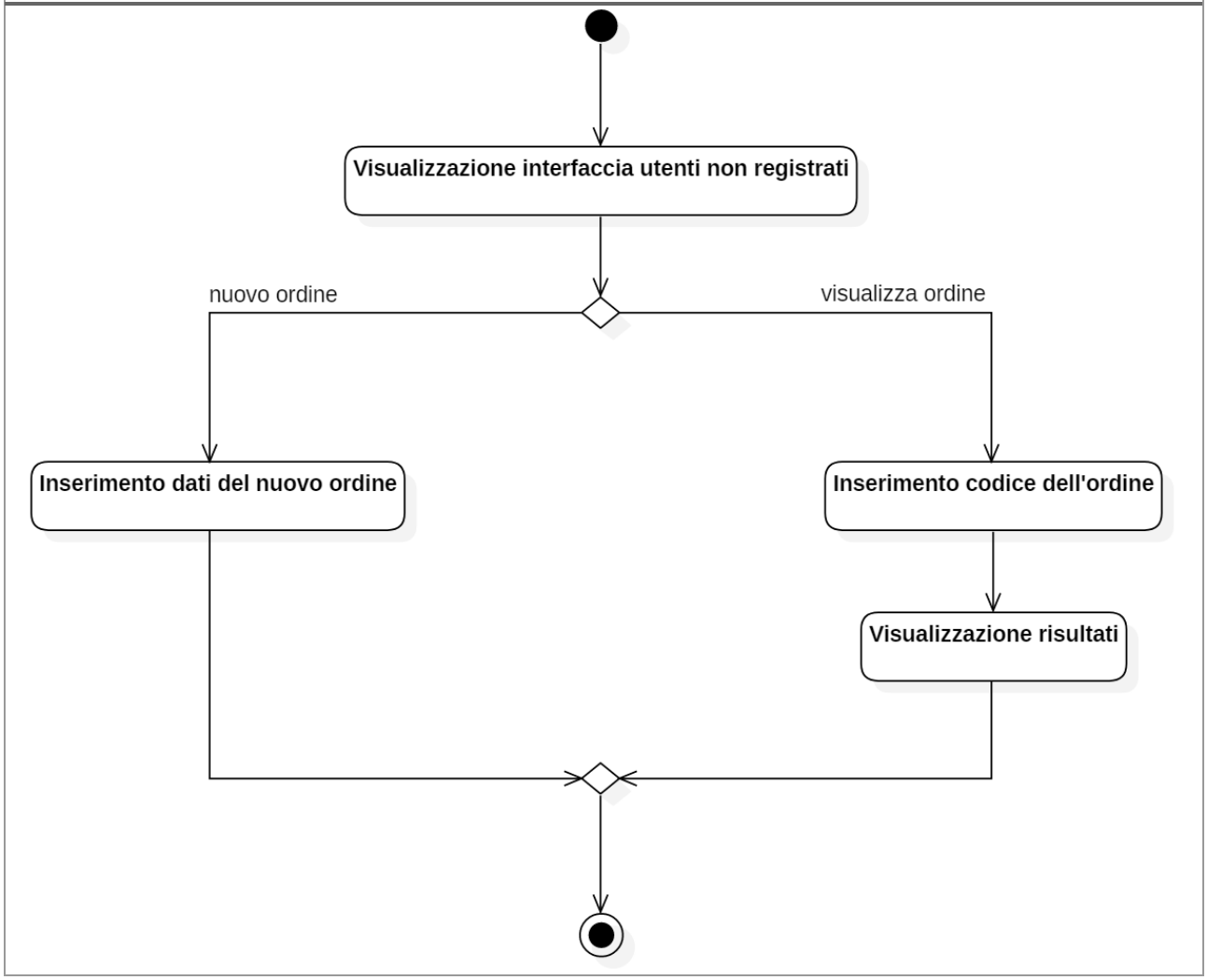
Attività: creazione di un ordine



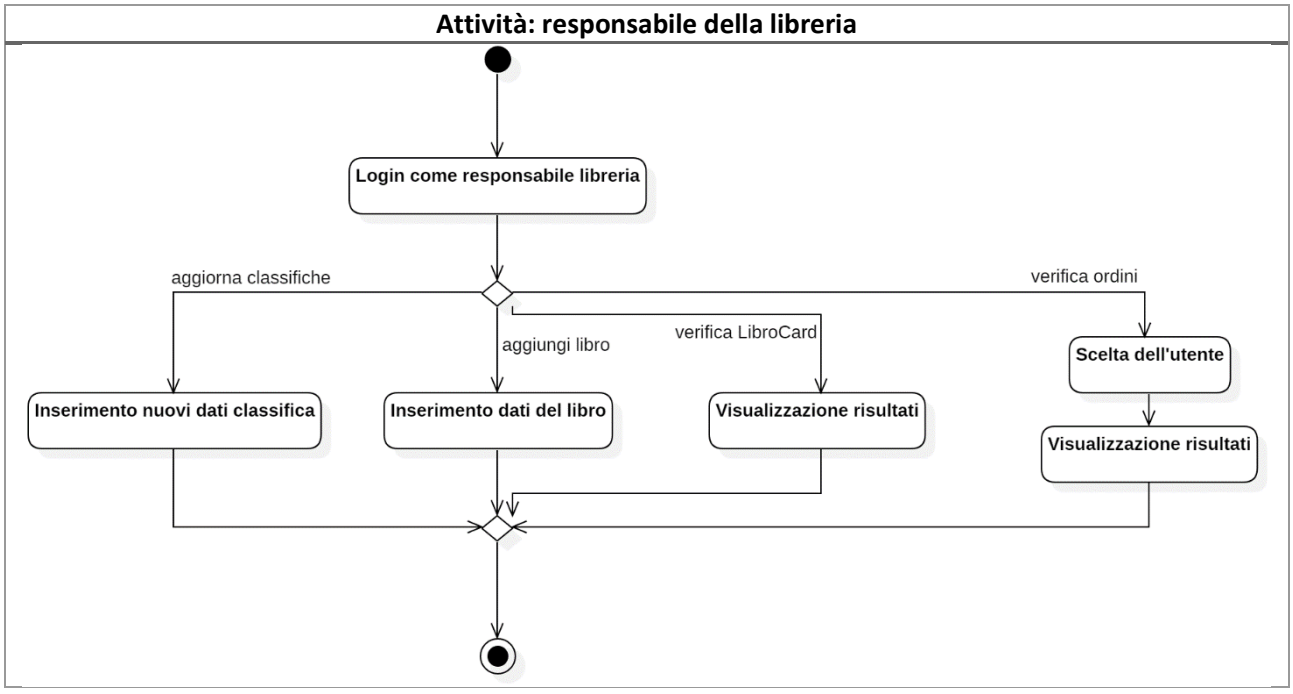
Attività: utente registrato



Attività: utente non registrato



Attività: responsabile della libreria

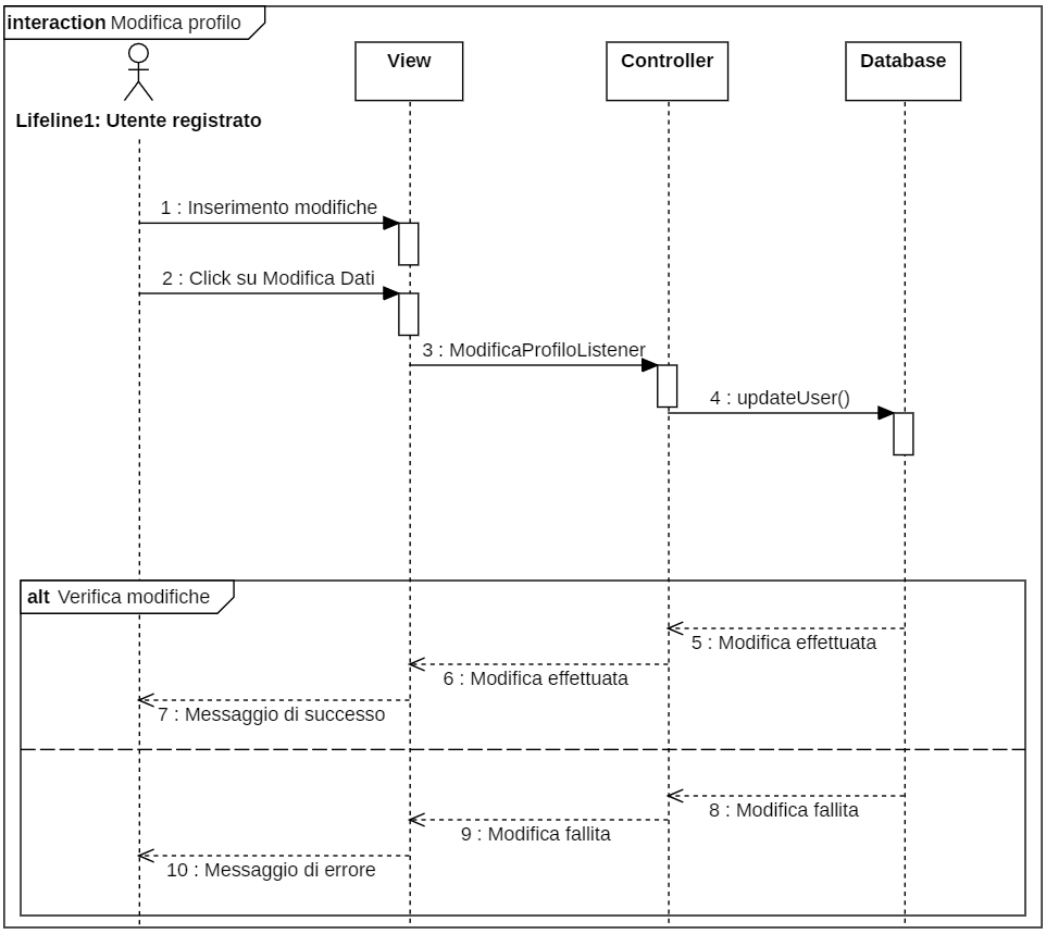


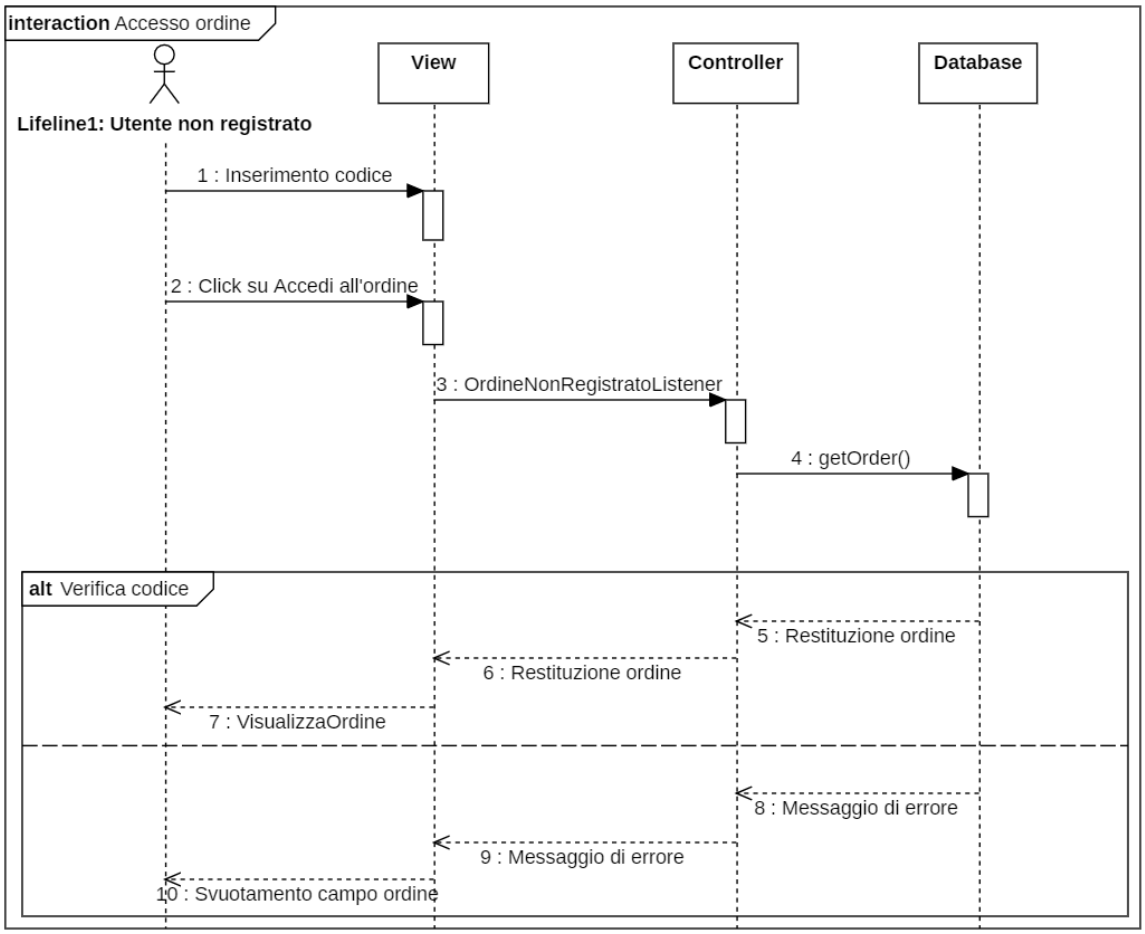
Diagrammi di sequenza

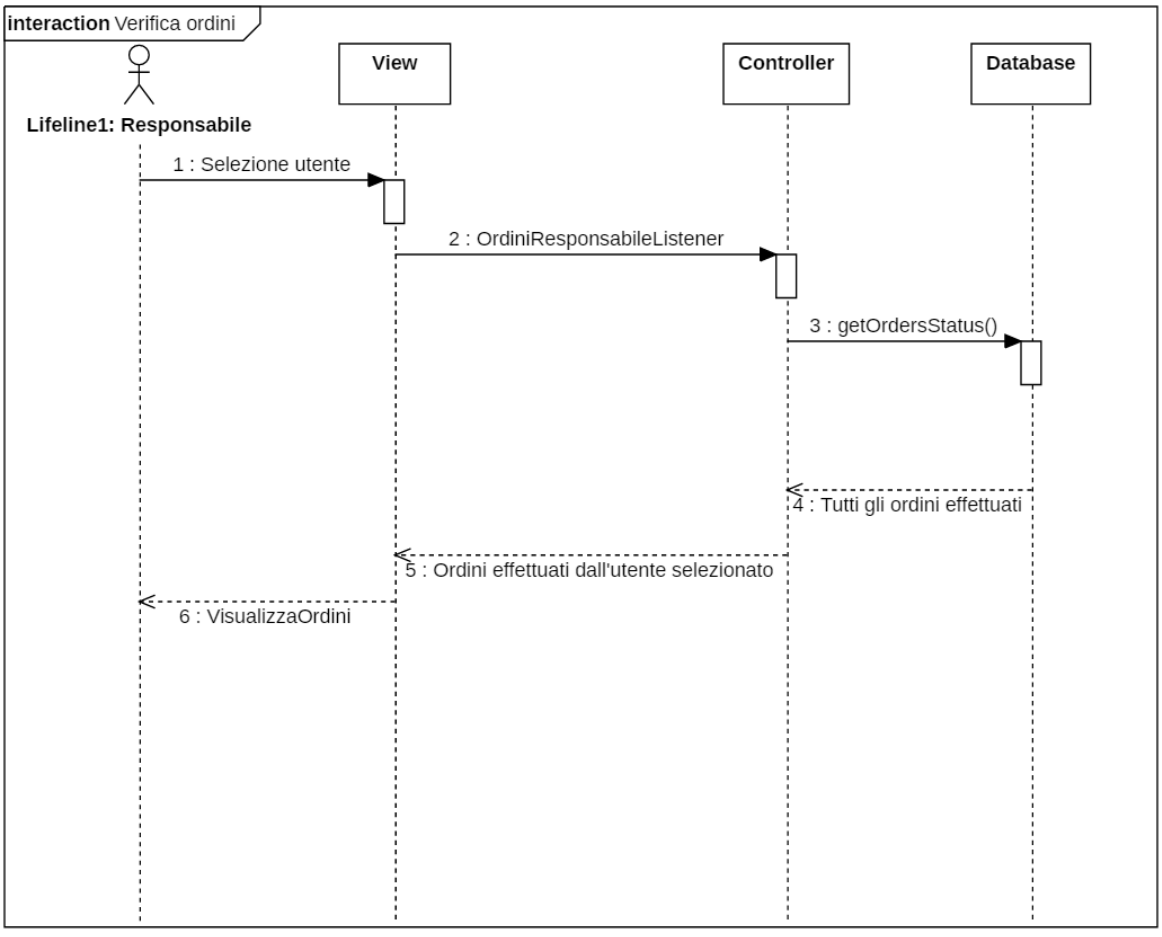
Note

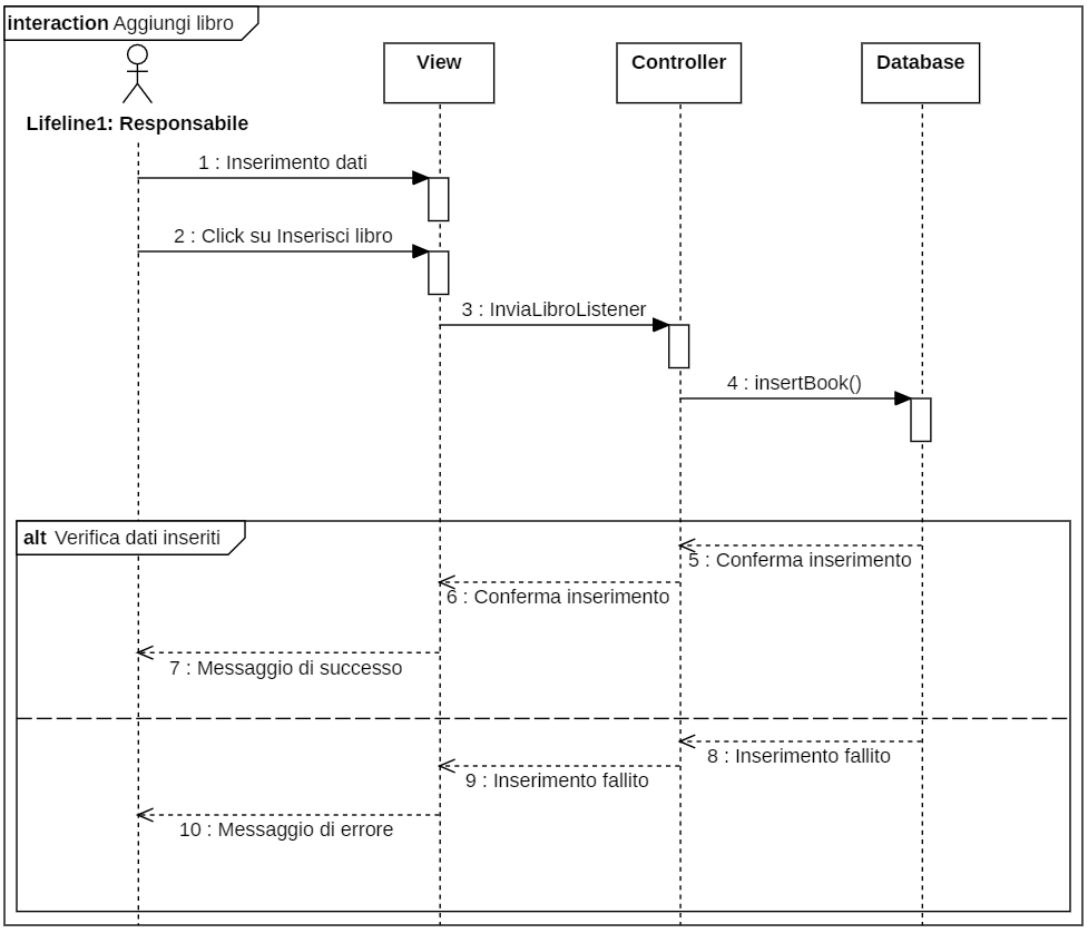
I *diagrammi di sequenza* modellano le interazioni (scambio di messaggi nel tempo) tra varie entità di un sistema. Il loro scopo è mostrare come un certo *comportamento* (un caso d'uso o un'operazione di classe) viene realizzato dalla collaborazione delle *entità* in gioco (attori o istanze di classe).

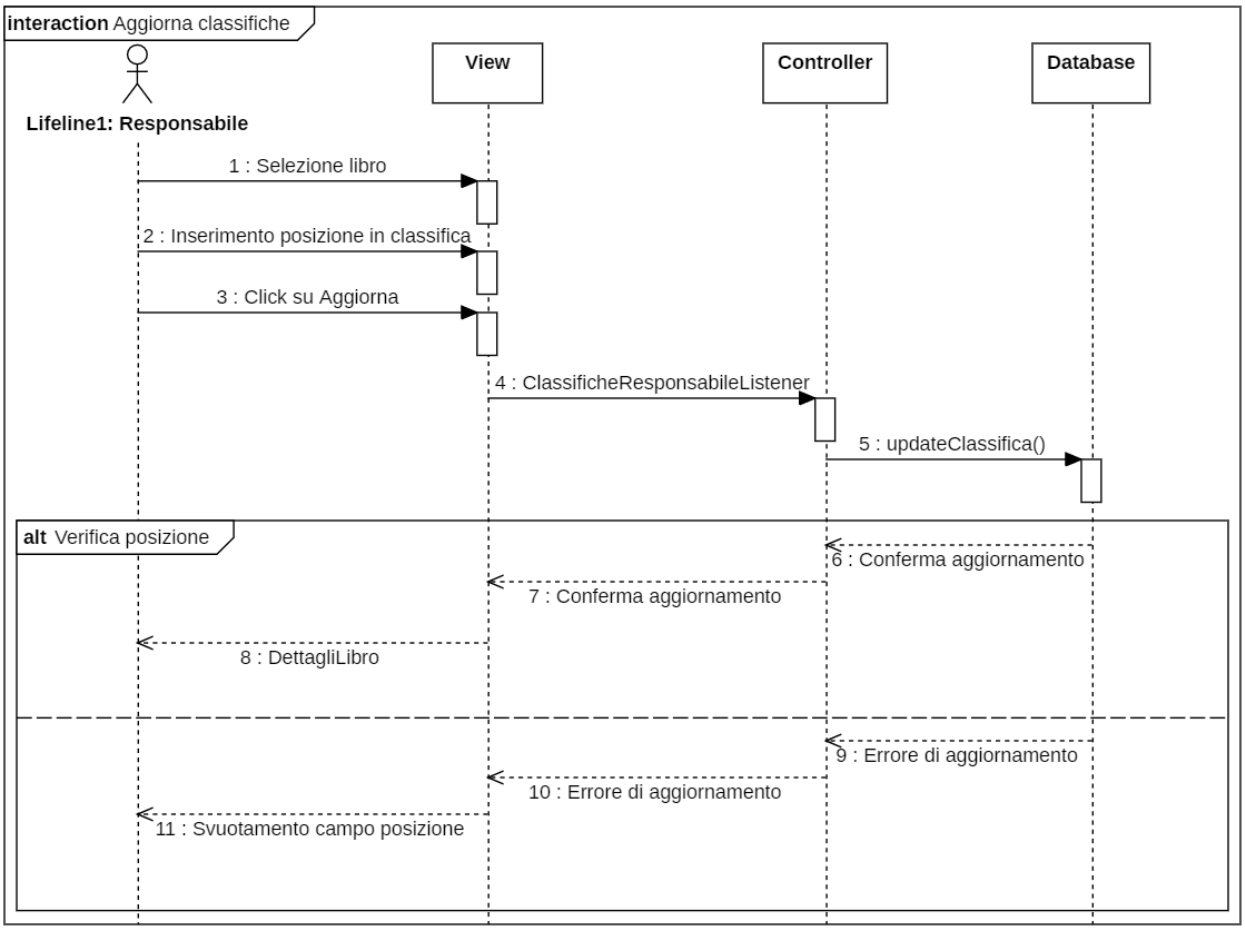
Diagrammi dei principali casi d'uso



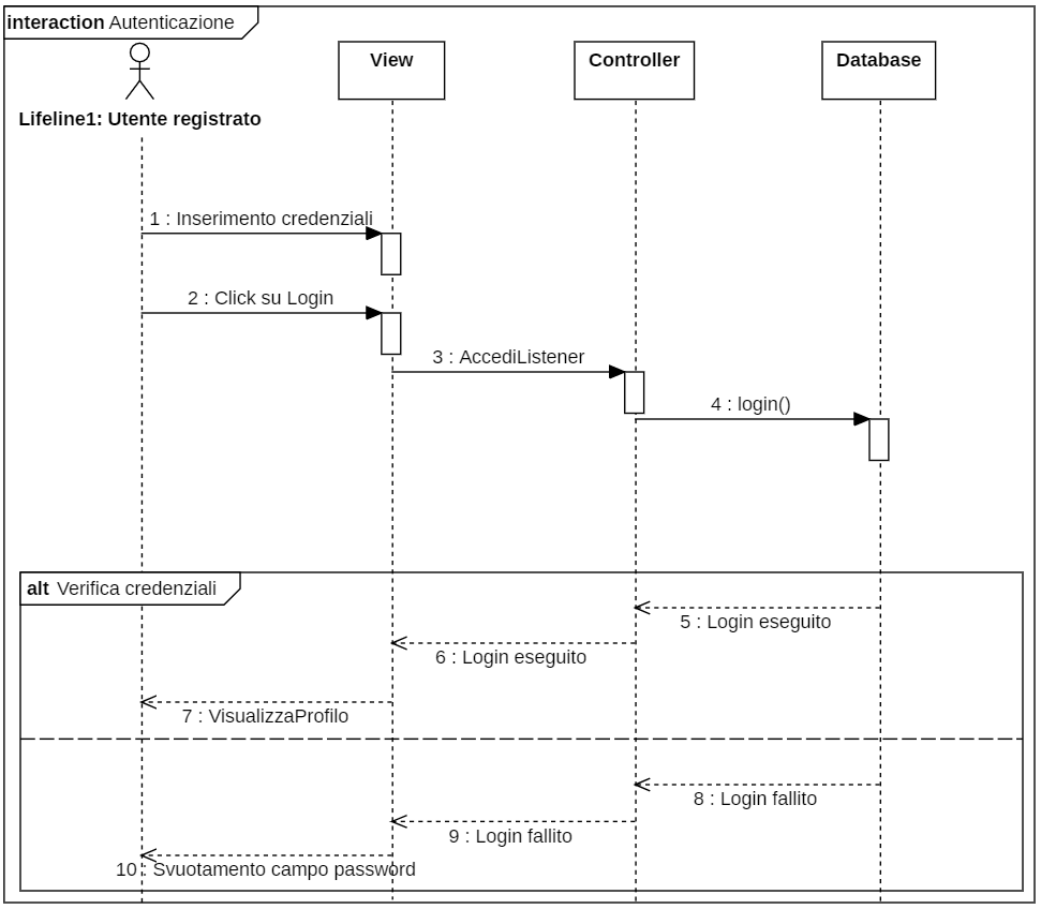

interaction Accesso ordine
View
Controller
Database
Lifeline1: Utente non registrato
1 : Inserimento codice
2 : Click su Accedi all'ordine
3 : OrdineNonRegistratoListener
4 : getOrder()
alt Verifica codice
5 : Restituzione ordine
6 : Restituzione ordine
7 : VisualizzaOrdine
8 : Messaggio di errore
9 : Messaggio di errore
10 : Svuotamento campo ordine

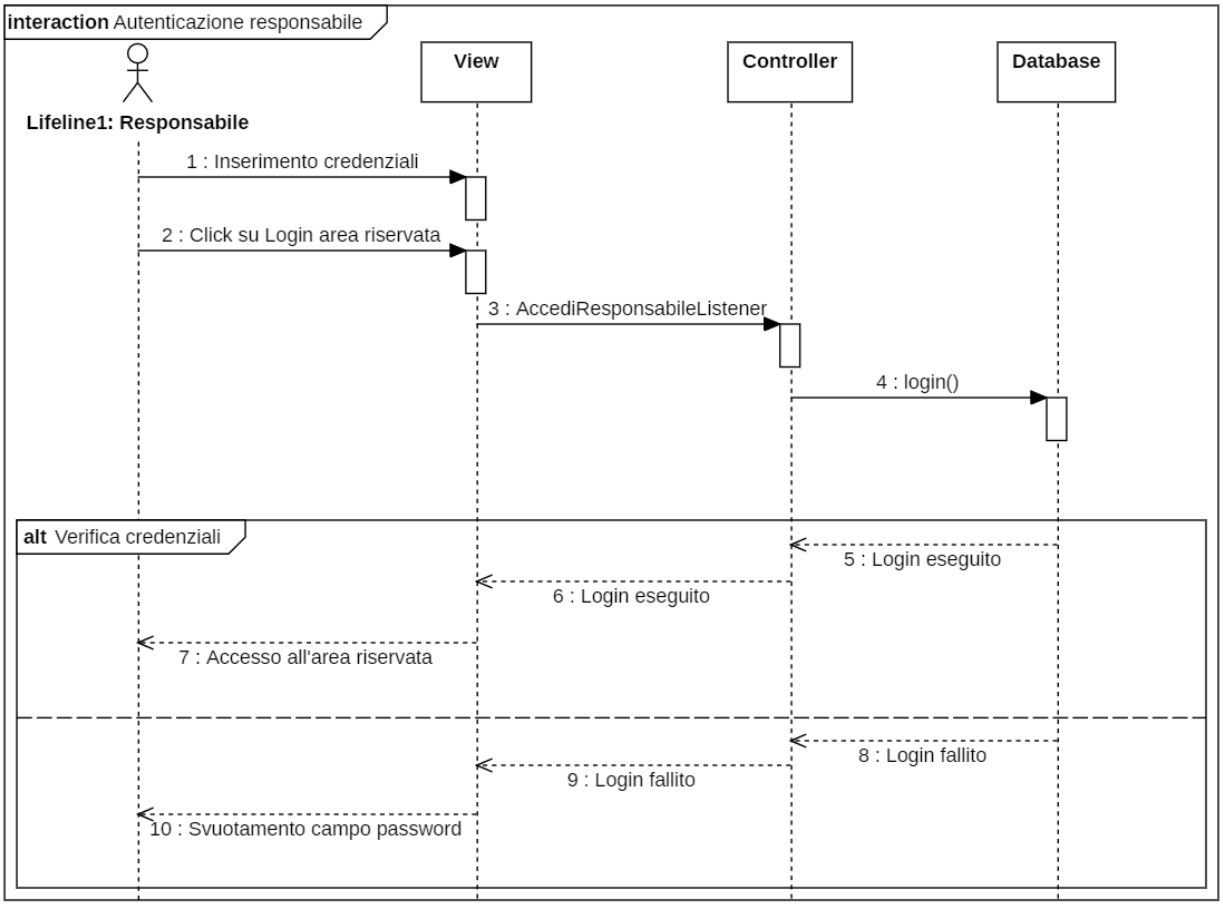

interaction Verifica ordini
View
Controller
Database
Lifeline1: Responsabile
1 : Selezione utente
2 : OrdiniResponsabileListener
3 : getOrdersStatus()
4 : Tutti gli ordini effettuati
5 : Ordini effettuati dall'utente selezionato
6 : VisualizzaOrdini

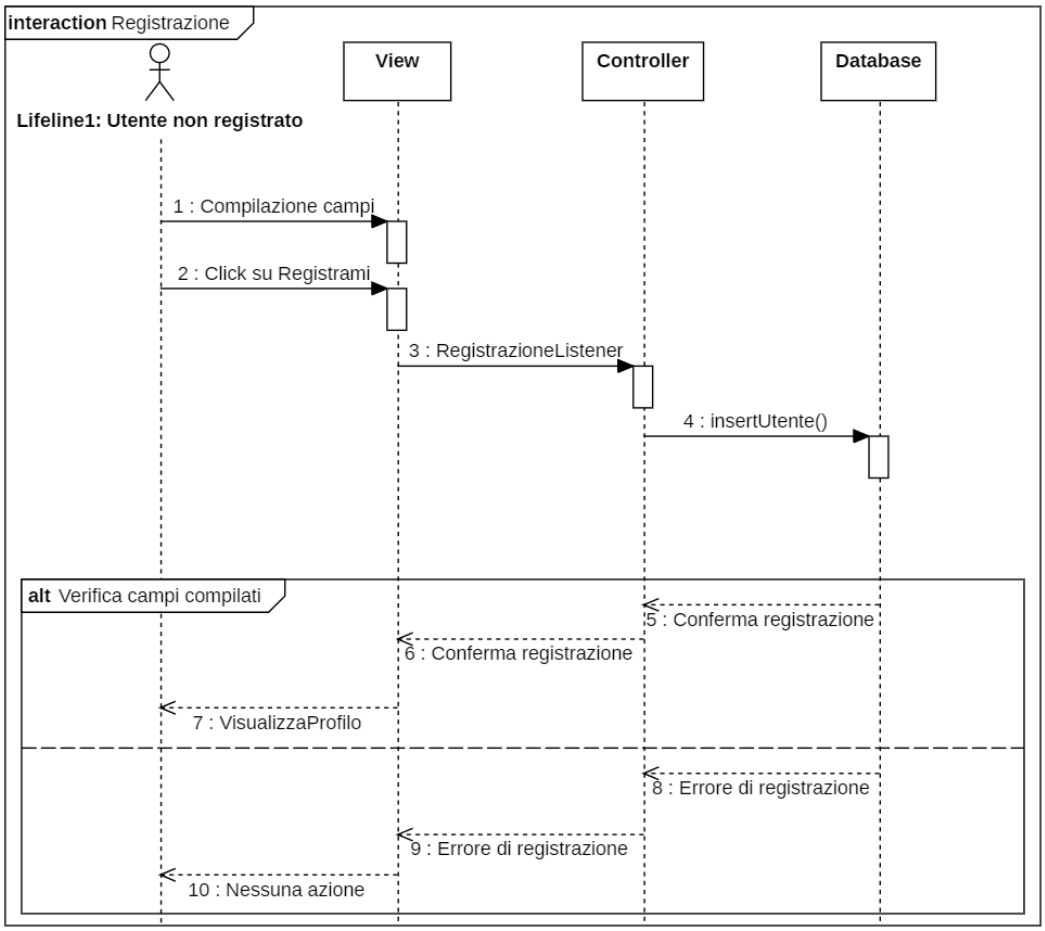

interaction Aggiungi libro
View
Controller
Database
Lifeline1: Responsabile
1 : Inserimento dati
2 : Click su Inserisci libro
3 : InviaLibroListener
4 : insertBook()
alt Verifica dati inseriti
5 : Conferma inserimento
6 : Conferma inserimento
7 : Messaggio di successo
8 : Inserimento fallito
9 : Inserimento fallito
10 : Messaggio di errore

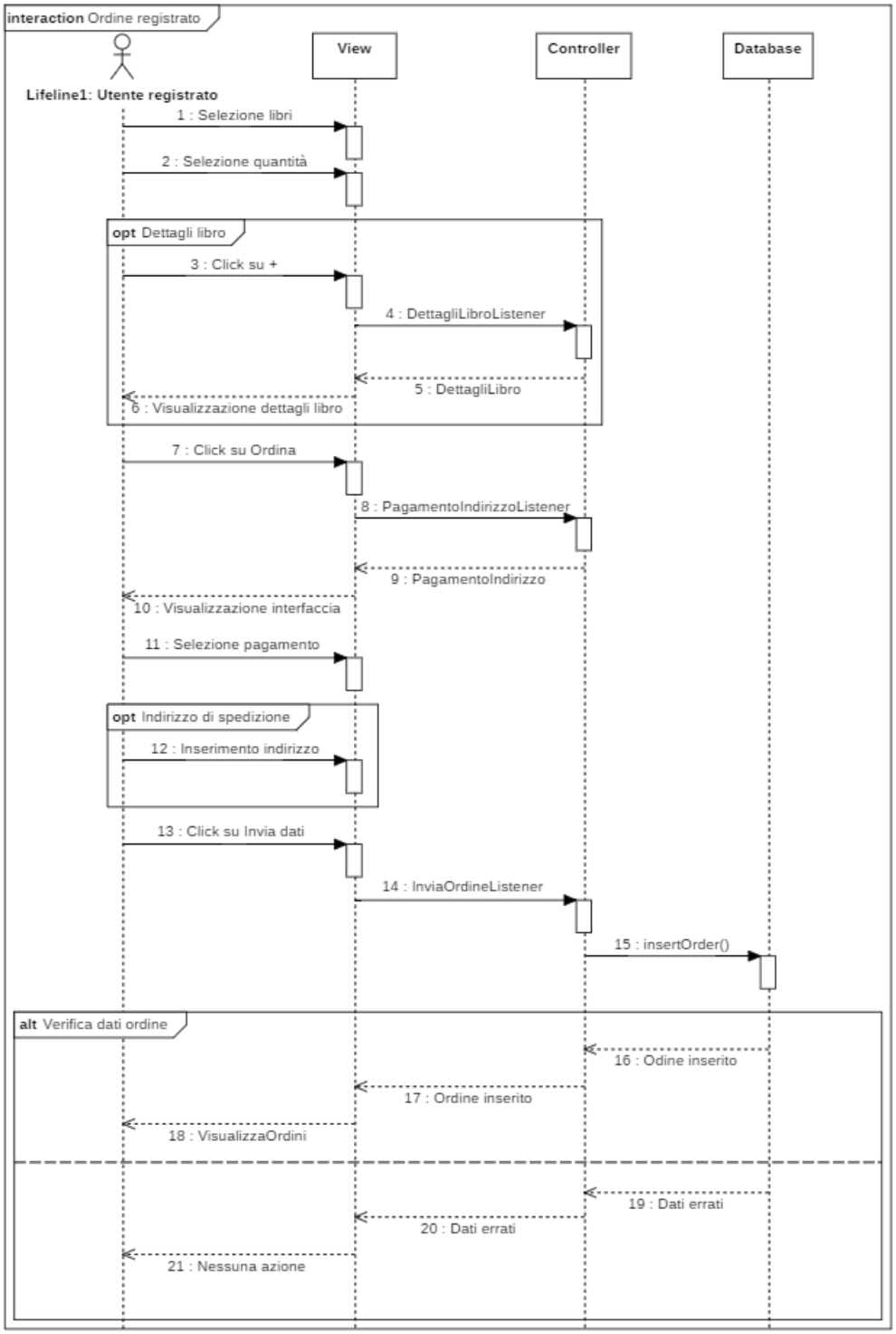

interaction Aggiorna classifiche
View
Controller
Database
Lifeline1: Responsabile
1 : Selezione libro
2 : Inserimento posizione in classifica
3 : Click su Aggiorna
4 : ClassificheResponsabileListener
5 : updateClassifica()
alt Verifica posizione
6 : Conferma aggiornamento
7 : Conferma aggiornamento
8 : DettagliLibro
9 : Errore di aggiornamento
10 : Errore di aggiornamento
11 : Svuotamento campo posizione

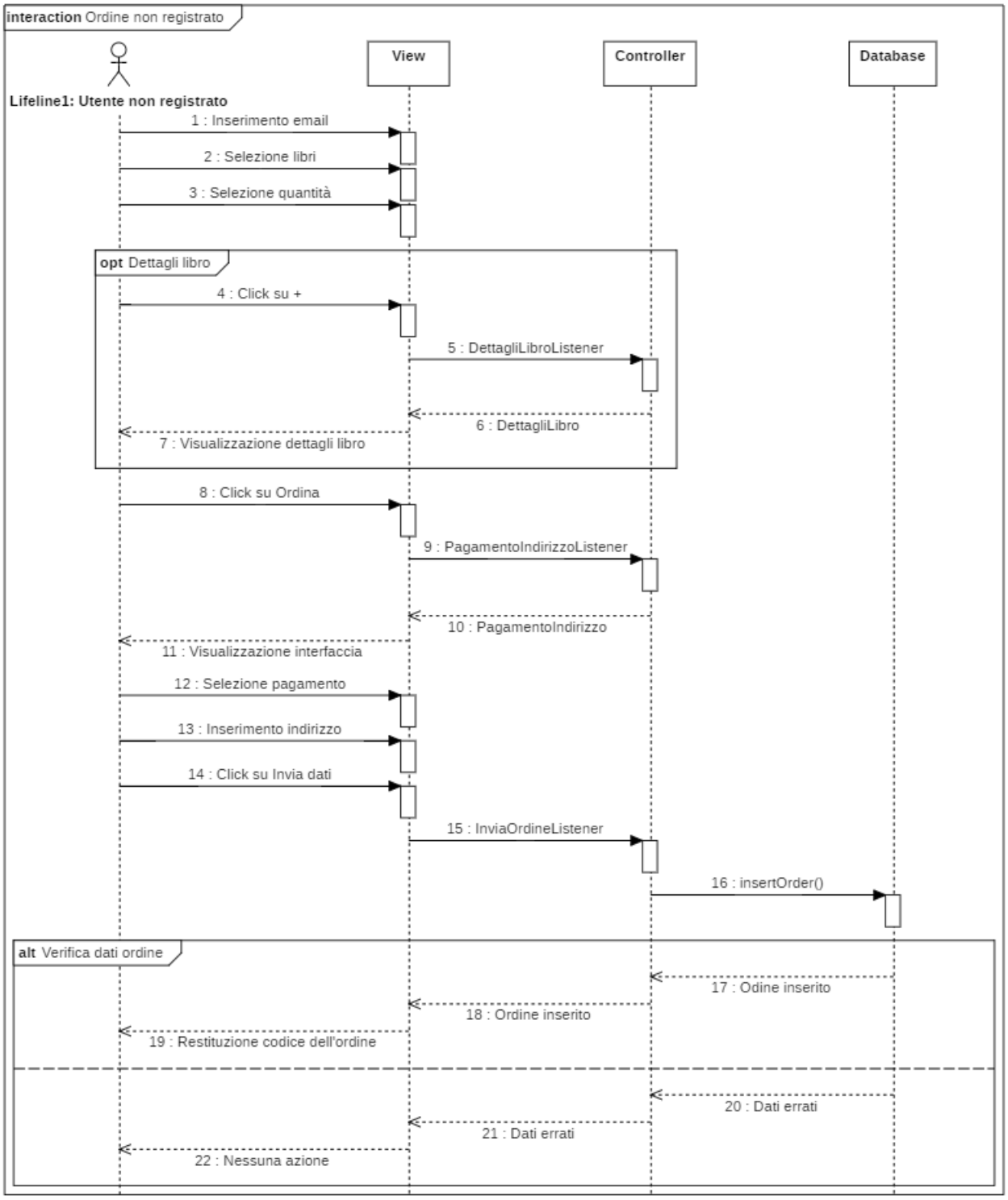
Diagrammi del software progettato






interaction Registrazione
View
Controller
Database
Lifeline1: Utente non registrato
1 : Compilazione campi
2 : Click su Registrami
3 : RegistrazioneListener
4 : insertUtente()
alt Verifica campi compilati
5 : Conferma registrazione
6 : Conferma registrazione
7 : VisualizzaProfilo
8 : Errore di registrazione
9 : Errore di registrazione
10 : Nessuna azione


interaction Ordine registrato
View
Controller
Database
Lifeline1: Utente registrato
1 : Selezione libri
2 : Selezione quantità
opt Dettagli libro
3 : Click su +
4 : DettagliLibroListener
5 : DettagliLibro
6 : Visualizzazione dettagli libro
7 : Click su Ordina
8 : PagamentoIndirizzoListener
9 : PagamentoIndirizzo
10 : Visualizzazione interfaccia
11 : Selezione pagamento
opt Indirizzo di spedizione
12 : Inserimento indirizzo
13 : Click su Invia dati
14 : InviaOrdineListener
15 : insertOrder()
alt Verifica dati ordine
16 : Odine inserito
17 : Ordine inserito
18 : VisualizzaOrdini
19 : Dati errati
20 : Dati errati
21 : Nessuna azione



Design pattern

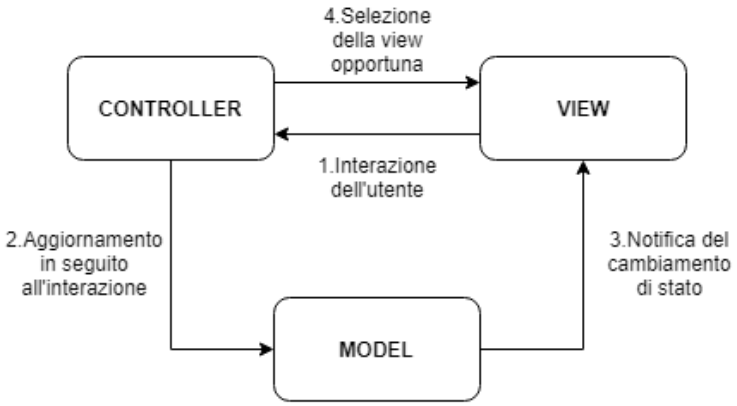
Note

I *pattern* sono un mezzo per rappresentare, condividere e riutilizzare le conoscenze software. Un *pattern architetturale* è una descrizione stilizzata di una buona pratica di progettazione che è stata provata e testata in diversi ambienti. I *pattern di progettazione*, invece, costituiscono degli schemi di base per progettare alcune porzioni di software con specifiche caratteristiche.

Model View Controller (MVC)

Per l'architettura di questo progetto è stato scelto di utilizzare il *pattern MVC*, che separa i moduli di presentazione e interazione dai moduli che contengono i dati. Il sistema è stato quindi strutturato in tre *package* contenenti le componenti logiche che interagiscono tra loro:

- Il package *Model* si occupa della gestione dei dati e delle operazioni che possono essere fatte su di essi; al suo interno sono presenti le classi che interagiscono con il database *Apache Derby* secondo il pattern DAO (vedere il paragrafo successivo per ulteriori dettagli).
- Il package *View* definisce l'interfaccia grafica e come i dati vengono presentati all'utente; il framework *Swing* utilizzato per produrre la suddetta interfaccia grafica implementa nativamente il pattern MVC.
- Il package *Controller* gestisce l'interazione dell'utente con l'interfaccia grafica elaborando le interazioni tra View e Model; al suo interno sono presenti i *listener* dell'interfaccia grafica, che realizzano le interazioni tra utente e sistema (vedere il paragrafo sui diagrammi di sequenza), e la classe *DaoManager* che crea la connessione con il database.



Data Access Object (DAO)

È un pattern architetturale per la gestione della *persistenza dei dati*, che consiste nel rappresentare un'entità tabellare di un database (Apache Derby nel caso di questo progetto) come una classe con relativi metodi. Lo scopo è quello di isolare l'accesso ad una tabella tramite query ponendo queste ultime all'interno dei metodi della classe, garantendo in questo modo un maggiore livello di *astrazione* ed una più facile *manutenibilità*. I metodi delle classi DAO, con le rispettive query al loro interno, verranno così richiamati dai listener dell'interfaccia grafica, mantenendo dunque una *rigida separazione* tra le componenti Model e Controller dell'applicazione. In particolare, per gestire ogni entità del database c'è bisogno di tre classi distinte:

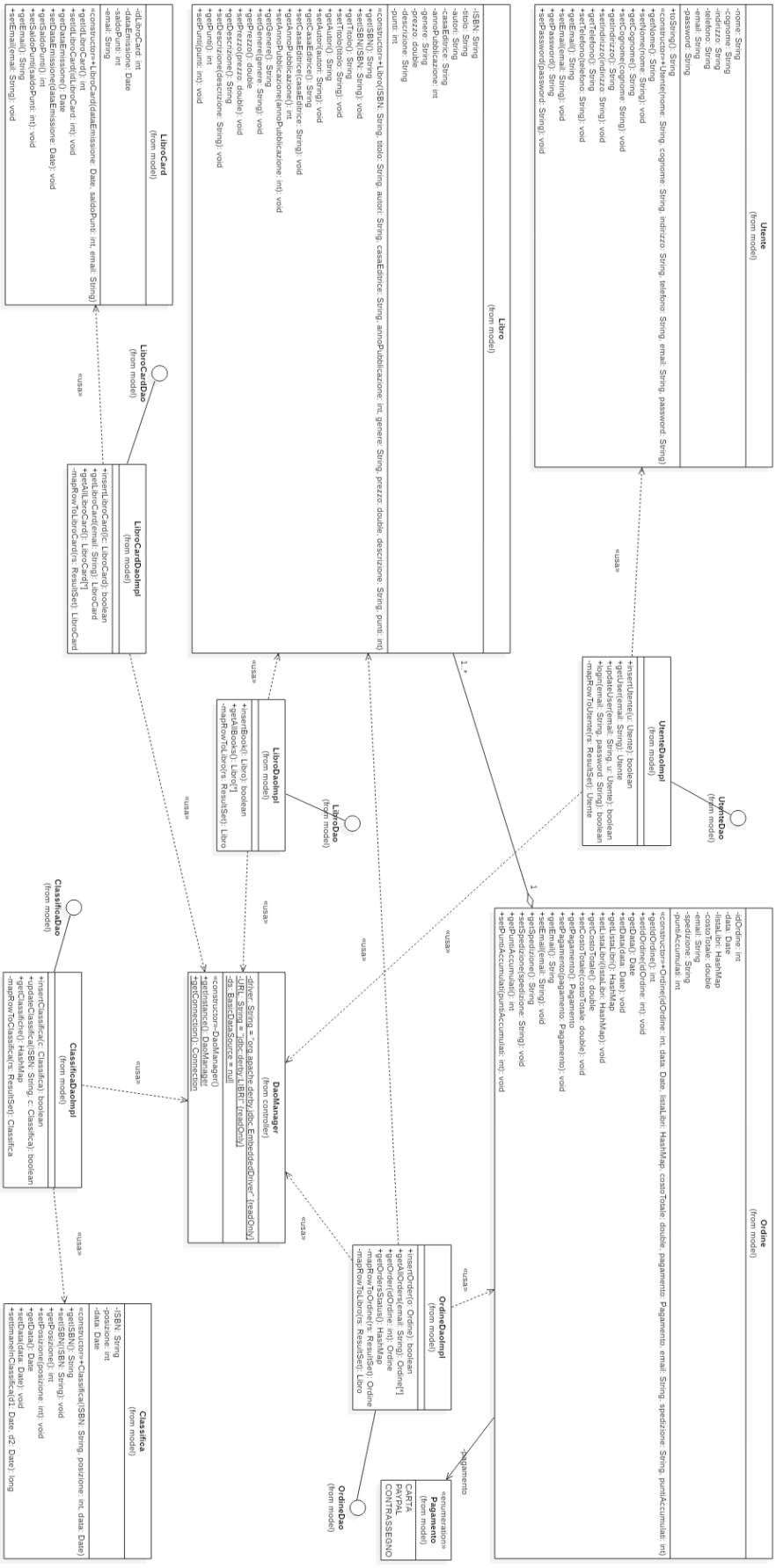
- Una classe che rappresenta l'entità del database come *oggetto* all'interno del *Model*.
- Un'*interfaccia DAO*, che definisce le operazioni che si possono fare sul database utilizzando l'oggetto corrispondente.
- Una *classe concreta* che rappresenta l'implementazione dell'interfaccia precedente la quale va ad inserire il codice per l'esecuzione delle query all'interno dei metodi definiti da quest'ultima.

Singleton

Il design pattern Singleton è usato per assicurare che una classe abbia *una sola istanza* per tutto il ciclo di vita del software. Le classi che rispettano questo design pattern hanno un *costruttore privato* (per evitare la possibilità di istanziare più oggetti della stessa classe) ed un *metodo statico*, il quale restituisce il riferimento all'unica istanza esistente (definita come campo statico). In questo progetto, due classi soddisfano il design pattern Singleton: la classe *DaoManager*, in quanto si vuole passare la stessa connessione al database a tutti gli oggetti che ne richiedono l'utilizzo (evitando di creare un nuovo oggetto ogni volta); e la classe *View*, in quanto contiene il frame principale dell'interfaccia grafica che può così essere usato da tutti gli altri panel.

Diagramma delle classi

Model



View e Controller

Attività di test

Descrizione

Durante la fase di test del software, usata per verificare che non ci siano difetti di sviluppo e quindi che il programma svolga quanto richiesto, sono state effettuate le seguenti attività:

- Ulteriore *analisi della specifica* e confronto con i diagrammi UML prodotti.
- *Test delle singole funzionalità del programma* per assicurarsi di avere un componente completamente funzionante prima di procedere allo sviluppo di altri componenti del programma.
- *Ispezione del codice* prodotto per verificare la consistenza con i diagrammi UML e la correttezza dei design pattern utilizzati.

Test dei componenti

Ogni volta che una nuova funzionalità del programma veniva sviluppata, sono stati effettuati dei test focalizzati principalmente sulle modalità di inserimento dei dati, per garantire la correttezza di questi ultimi e ridurre i danni che verrebbero causati dagli errori degli utenti generici, ovvero quelli non coinvolti nello sviluppo e quindi privi di una particolare dimestichezza informatica. Nello specifico, sono stati testati i seguenti componenti:

- *Autenticazione*, verificando che dalla pagina di autenticazione sia possibile eseguire l'accesso (esclusivamente attraverso e-mail e password fornite in fase di registrazione) per visualizzare le pagine relative alle operazioni che può svolgere l'utente registrato.
- *Registrazione*, verificando che nella pagina di registrazione sia possibile inserire in maniera corretta tutti i dati necessari a registrare un nuovo utente.
- *Creazione di un ordine*, verificando che gli utenti possano inserire in maniera corretta tutti i dati necessari alla creazione di un nuovo ordine.
- *Modifica profilo*, verificando che nella pagina per la modifica del profilo di un utente registrato sia possibile inserire in maniera corretta tutti i dati che si vogliono cambiare.
- *Non registrati*, verificando che gli utenti non registrati siano in grado di visualizzare le pagine relative alle operazioni che possono svolgere.
- *Accesso tramite codice ordine*, verificando che gli utenti non registrati siano in grado di inserire in maniera corretta il codice dell'ordine che vogliono visualizzare.
- *Area riservata*, verificando che dalla pagina di autenticazione dei responsabili della libreria sia possibile eseguire l'accesso (esclusivamente attraverso username e password definiti a priori) per visualizzare le pagine relative alle operazioni che possono svolgere.
- *Inserimento libro*, verificando che i responsabili della libreria possano inserire in maniera corretta tutti i dati necessari all'aggiunta di un nuovo libro al catalogo.
- *Aggiornamento classifiche*, verificando che i responsabili della libreria possano inserire in maniera corretta tutti i dati necessari ad aggiornare la posizione in classifica di un libro nel catalogo.