



Università degli Studi di Verona

Ingegneria del Software

Gestione Magazzino Articoli Sportivi

Documentazione prototipo

Nicoló Beato, Filippo De Guio, Emanuele Lazzaretti

Luglio 2018

Indice

1. <u>Introduzione</u>	3
2. <u>Descrizione dei possibili errori previsti</u>	4
3. <u>UML</u>	5
<u>3.1 Casi d'uso</u>	5
<u>3.2 Diagrammi delle attività</u>	11
<u>3.3 Diagramma delle sequenze</u>	16
<u>3.3 Diagramma delle classi</u>	28
<u>3.4 Diagramma dei package</u>	30
4. <u>Scelte Progettuali</u>	31
<u>4.1 Sviluppo</u>	31
<u>4.2 Database</u>	31
<u>4.3 Organizzazione della GUI</u>	32
<u>4.4 MVC Pattern</u>	34
<u>4.5 Singleton Pattern</u>	34
<u>4.6 Facade Pattern</u>	34
5. <u>Test e validazione</u>	34

1 Introduzione ed obiettivi

Obiettivo dell'elaborato è la progettazione di un prototipo software per la gestione del magazzino di una catena di negozi di articoli sportivi.

In questa relazione ci proponiamo di raccogliere la documentazione sviluppata e di fornire spiegazioni e dettagli sulle scelte progettuali ed implementative.

Requisiti

Di seguito viene riportato il testo dell'elaborato.

Si vuole progettare un sistema informatico per gestire il magazzino di una catena di negozi di articoli sportivi.

Il negozio vende articoli di diversa tipologia, raggruppati per sport. Per ogni tipo articolo si registra: un nome univoco, una descrizione, lo sport, e i materiali utilizzati per produrlo. Il sistema registra tutti gli articoli in magazzino memorizzando per ogni articolo: il tipo di articolo, un codice univoco, il prezzo e la data di produzione.

Gli articoli in magazzino vengono gestiti dal sistema che registra per ogni ingresso in magazzino: un codice interno univoco, la data e tutti articoli entrati e le loro posizioni in magazzino. Per ogni uscita il sistema registra: la data e il numero di bolla (univoco), tutti gli articoli usciti, il negozio che li ha ordinati e lo spedizioniere che li ritira. Per ogni negozio della catena il sistema registra: il codice fiscale, il nome, l'indirizzo e la città.

Il sistema memorizza inoltre gli ordini dei negozi registrando: il negozio che ha effettuato l'ordine, un codice ordine univoco, la data dell'ordine, i tipi di articolo ordinati e per ogni tipo di articolo la quantità ordinata e il prezzo totale. Quando un ordine viene evaso si registra un'uscita dal magazzino che viene collegata all'ordine al quale si riferisce. Si suppone che per ogni ordine evaso si abbia una sola uscita dal magazzino.

Per ogni tipo di articolo il sistema memorizza esplicitamente alla fine di ogni mese dell'anno la quantità di articoli ricevuti in magazzino e la quantità di articoli usciti.

Il sistema deve permettere ai magazzinieri di inserire le informazioni relative ai movimenti di ingresso e uscita dal magazzino. I magazzinieri, inoltre, possono spostare un articolo da una posizione ad un'altra del magazzino, al fine di ottimizzare l'occupazione del magazzino.

La segreteria amministrativa della catena di negozi è responsabile dell'inserimento dei tipi di articolo. Essa può accedere al sistema e visualizzare i movimenti di magazzino rispetto agli ordini dei vari negozi. Tutti gli utenti sono opportunamente autenticati dal sistema, prima che possano accedere alle funzionalità specifiche. I responsabili dei negozi possono accedere al sistema per effettuare gli ordini e per avere un riassunto degli ordini passati.

2 Descrizione dei possibili errori previsti

PROBLEMA: credenziali di login errate

SOLUZIONE:

- Viene visualizzato un messaggio di errore ed il login non viene effettuato

PROBLEMA: l'utente prova a confermare un ordine mentre il carrello è ancora vuoto

SOLUZIONE:

- Viene visualizzato un messaggio di errore e l'ordine non viene confermato

PROBLEMA: l'utente mentre effettua l'inserimento della quantità di prodotti da ordinare prova ad inserire un carattere diverso da un numero

SOLUZIONE:

- L'utente non è in grado di inserire l'articolo selezionato nel carrello

PROBLEMA: durante lo spostamento di un articolo all'interno del magazzino viene inserito un codice prodotto errato

SOLUZIONE:

- Viene visualizzato un messaggio di errore e lo spostamento non viene effettuato

PROBLEMA: durante l'inserimento di un articolo nel magazzino non vengono completati correttamente tutti i campi di testo

SOLUZIONE:

- Viene visualizzato un messaggio di errore che invita l'utente a completare correttamente tutti i campi di testo

PROBLEMA: durante l'inserimento di un tipo articolo viene inserito un nome già in uso nel sistema

SOLUZIONE:

- Viene visualizzato un messaggio di errore che avvisa l'utente che il nome scelto è già presente nel sistema

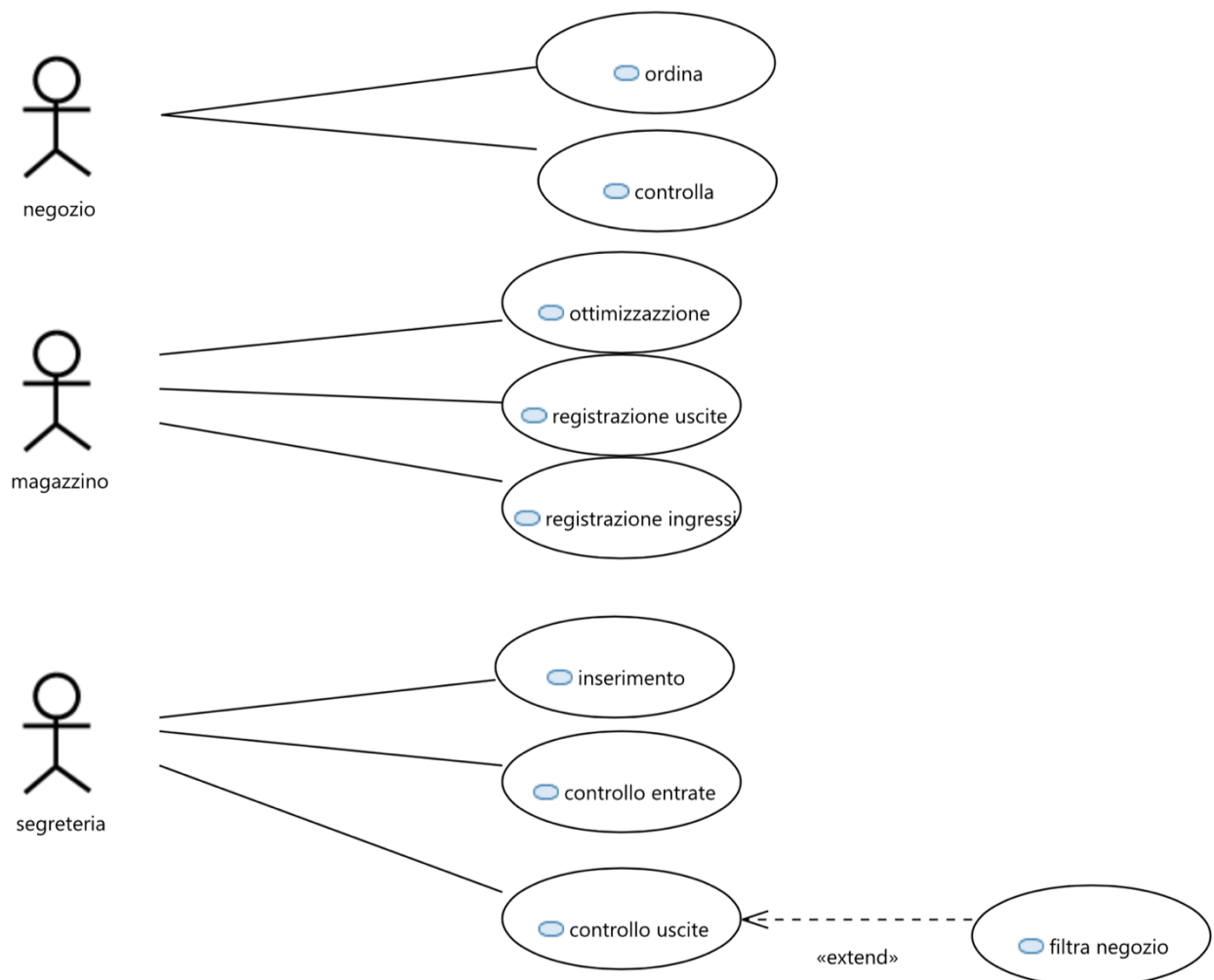
PROBLEMA: durante l'inserimento di un tipo articolo non vengono completati correttamente tutti i campi di testo

SOLUZIONE:

- Viene visualizzato un messaggio di errore che invita l'utente a completare correttamente tutti i campi di testo

3.1 Casi d'uso

In questa sezione sono presentati i casi d'uso richiesti per la documentazione del prototipo e le relative sequenze semplificate.



1. CASO D'USO: Ordina

Id: CU1

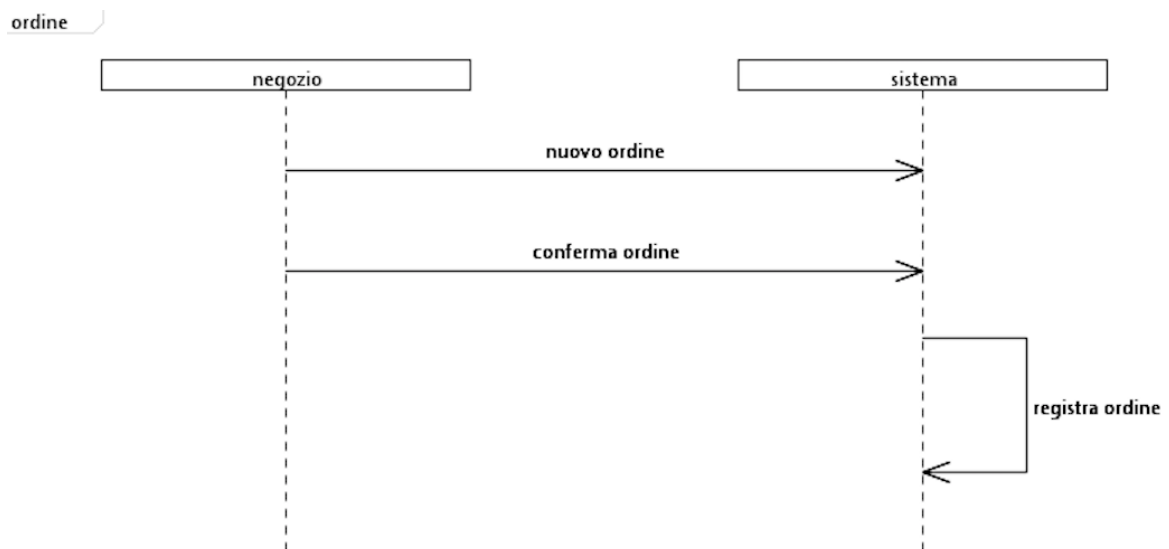
Attori: Negozio

Precondizione: è stato eseguito con successo il login come negozio

Sequenza degli eventi:

1. il caso d'uso inizia quando il negozio entra nella scheda nuovo ordine
2. il negozio inserisce la quantità e gli articoli
3. il negozio conferma l'ordine

Postcondizione: ordine registrato dal sistema



2. CASI D'USO: Controllo ordini passati

Id: CU2

Attori: Negozio

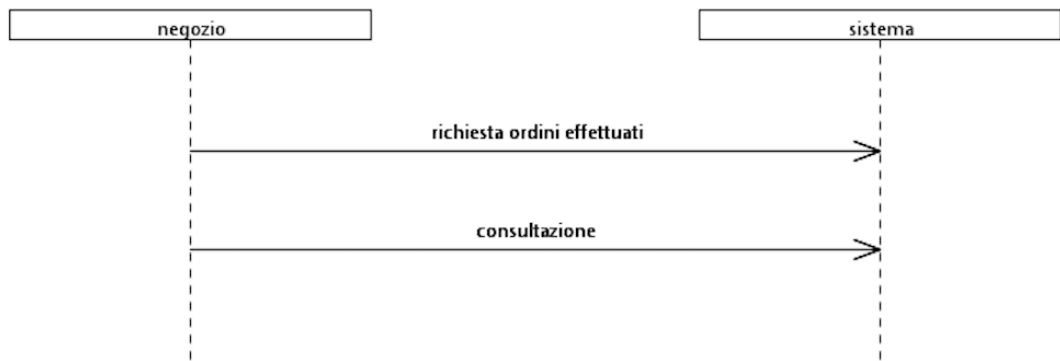
Precondizioni: è stato eseguito con successo il login come negozio

Sequenza degli eventi:

1. Il caso d'uso inizia quando il negozio entra nella scheda ordini effettuati
2. Il negozio visualizza lo storico dei suoi ordini e può consultarli

Postcondizione: nessuna

controlla ordini



3. CASO D'USO: registrazione ingressi articolo

Id: CU3

Attori: Magazzino

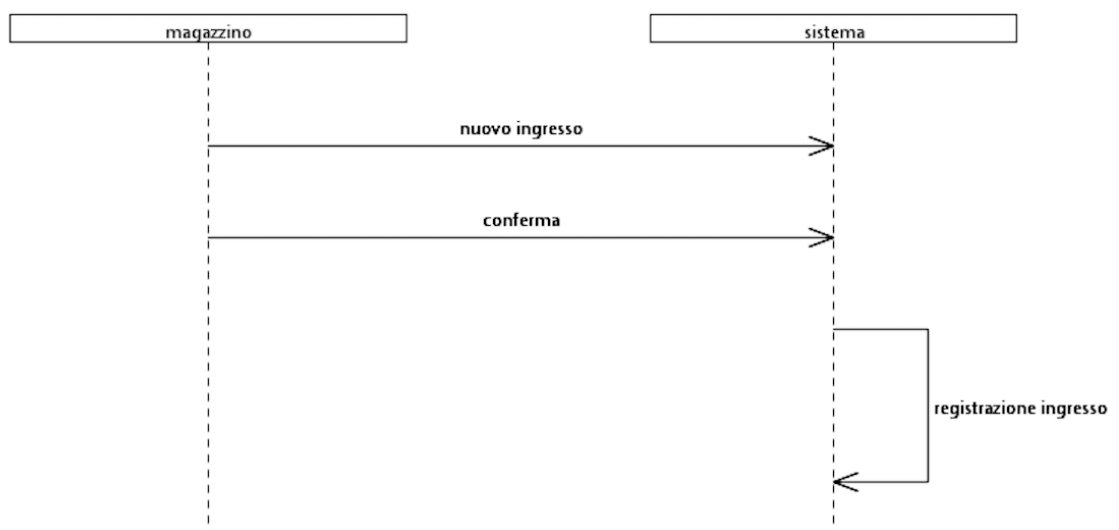
Precondizione: è stato eseguito con successo il login come magazzino

Sequenza degli eventi:

1. il caso d'uso inizia quando il magazzino entra sulla scheda nuovo ingresso
2. il magazzino sceglie il tipo articolo, inserisce la posizione, data produzione e il prezzo
3. una volta inseriti gli articoli non resta che confermare l'ingresso

Postcondizione: il sistema registra un ingresso di articoli

registrazione ingresso



4. CASO D'USO: *registrazione uscita articolo*

Id: CU4

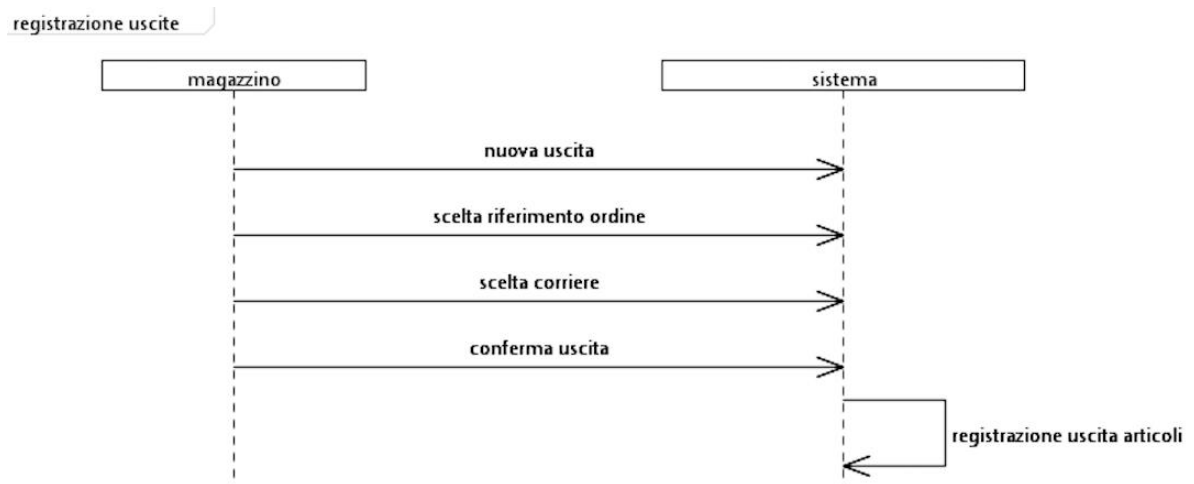
Attori: Magazzino

Precondizione: è stato eseguito con successo il login come magazzino

Sequenza degli eventi:

1. il caso d'usa inizia quando il magazzino entra sulla scheda nuova uscita
2. il magazzino sceglie il riferimento dell'ordine e lo spedizioniere
3. il magazzino visualizza la lista delle giacenze presenti per soddisfare le richieste dell'ordine
4. aggiunge gli articoli alla lista dell'uscita e conferma

Postcondizione: il sistema registra un'uscita di articoli



5. CASO D'USO: *ottimizzazione*

Id: CU5

Attori: Magazzino

Precondizione: è stato eseguito con successo il login come magazzino

Sequenza degli eventi:

1. il caso d'uso inizia quando il magazzino entra sulla scheda spostamento
2. il magazzino inserisce il codice articolo, inserisce la nuova posizione e clicca su sposta

Postcondizione: lo spostamento viene registrato dal sistema

ottimizzazione



6. CASO D'USO: *inserimento tipo articolo*

Id: CU6

Attori: Segreteria

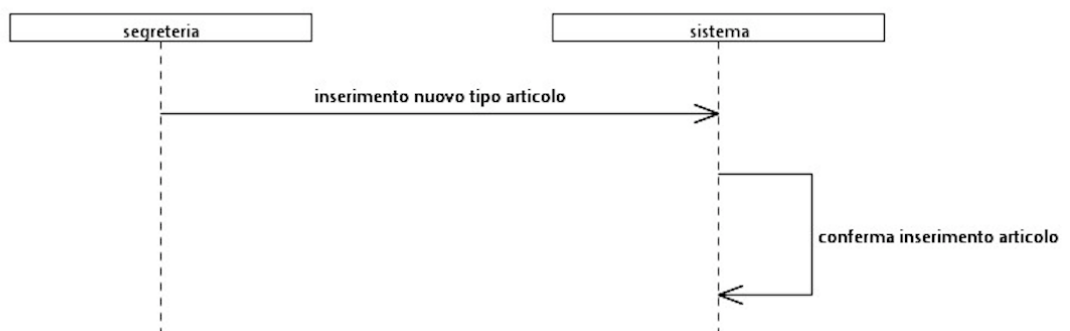
Precondizione: è stato eseguito con successo il login come segreteria

Sequenza degli eventi:

1. il caso d'uso inizia quando la segreteria entra nella scheda inserimento tipo articolo
2. la segreteria deve compilare il form inserendo nome, descrizione, sport e materiali
3. la segreteria una volta compilato correttamente il form, conferma l'inserimento

Postcondizione: il sistema registra l'inserimento di un nuovo tipo articolo

inserimento tipo articolo



7. CASO D'USO: controllo movimenti magazzino entrata

Id: CU7

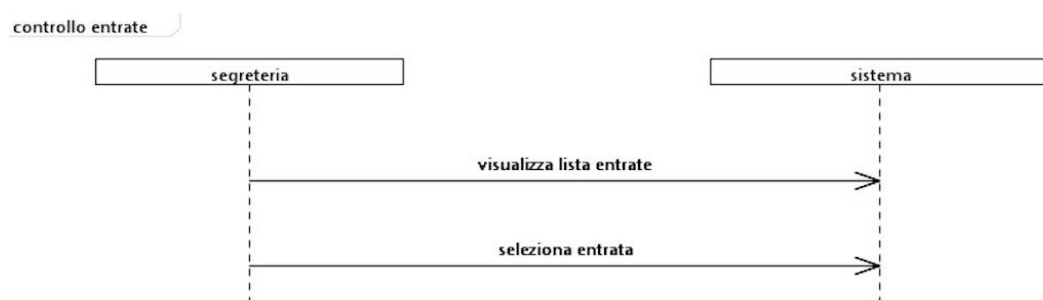
Attori: Segreteria

Precondizione: è stato eseguito con successo il login come segreteria

Sequenza degli eventi:

1. il caso d'uso inizia quando la segreteria entra sulla scheda entrata
2. la segreteria visualizza la lista delle entrate e, selezionandone una, ne visualizza il dettaglio

Postcondizione: nessuna



8. CASO D'USO: controllo movimenti magazzino uscita

Id: CU8

Attori: Segreteria

Precondizione: è stato eseguito con successo il login come segreteria

Sequenza degli eventi:

1. il caso d'uso inizia quando la segreteria entra nella scheda uscite
2. la segreteria può vedere la lista di tutte le uscite
 - 2.1. la segreteria filtra le uscite scegliendo il codice fiscale di un negozio
3. la segreteria cliccando su dettaglio vede il dettaglio dell'uscita con il relativo ordine

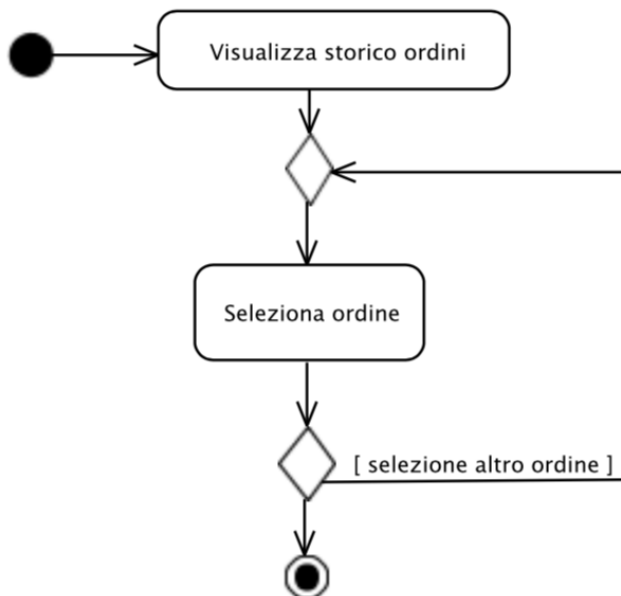
Postcondizione: nessuna



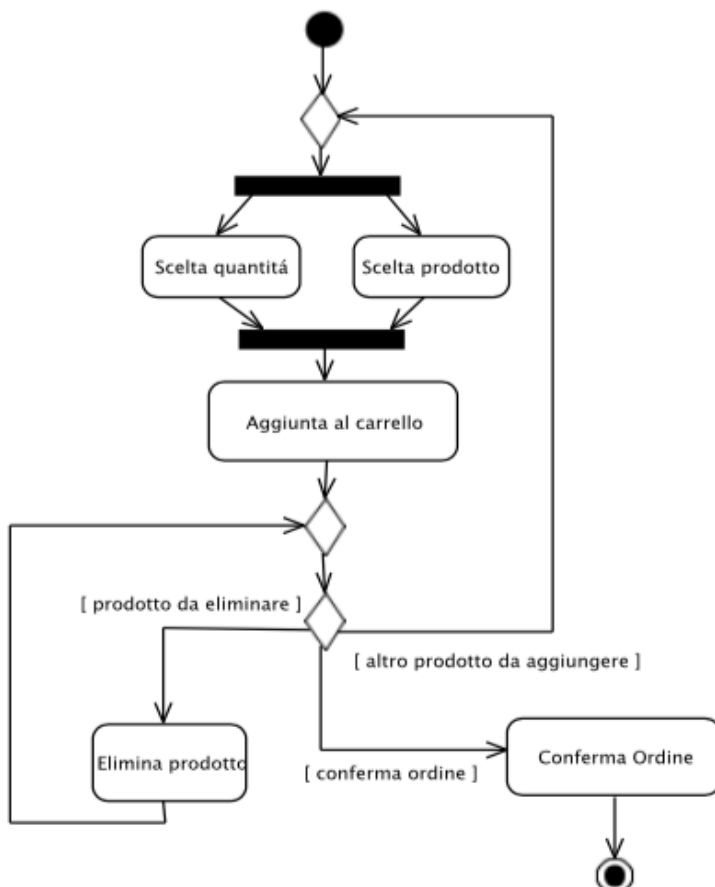
3.2 Diagrammi delle attività

I seguenti diagrammi delle attività cercano di modellare in modo più dettagliato il flusso d'esecuzione relativo agli use-case precedentemente proposti.

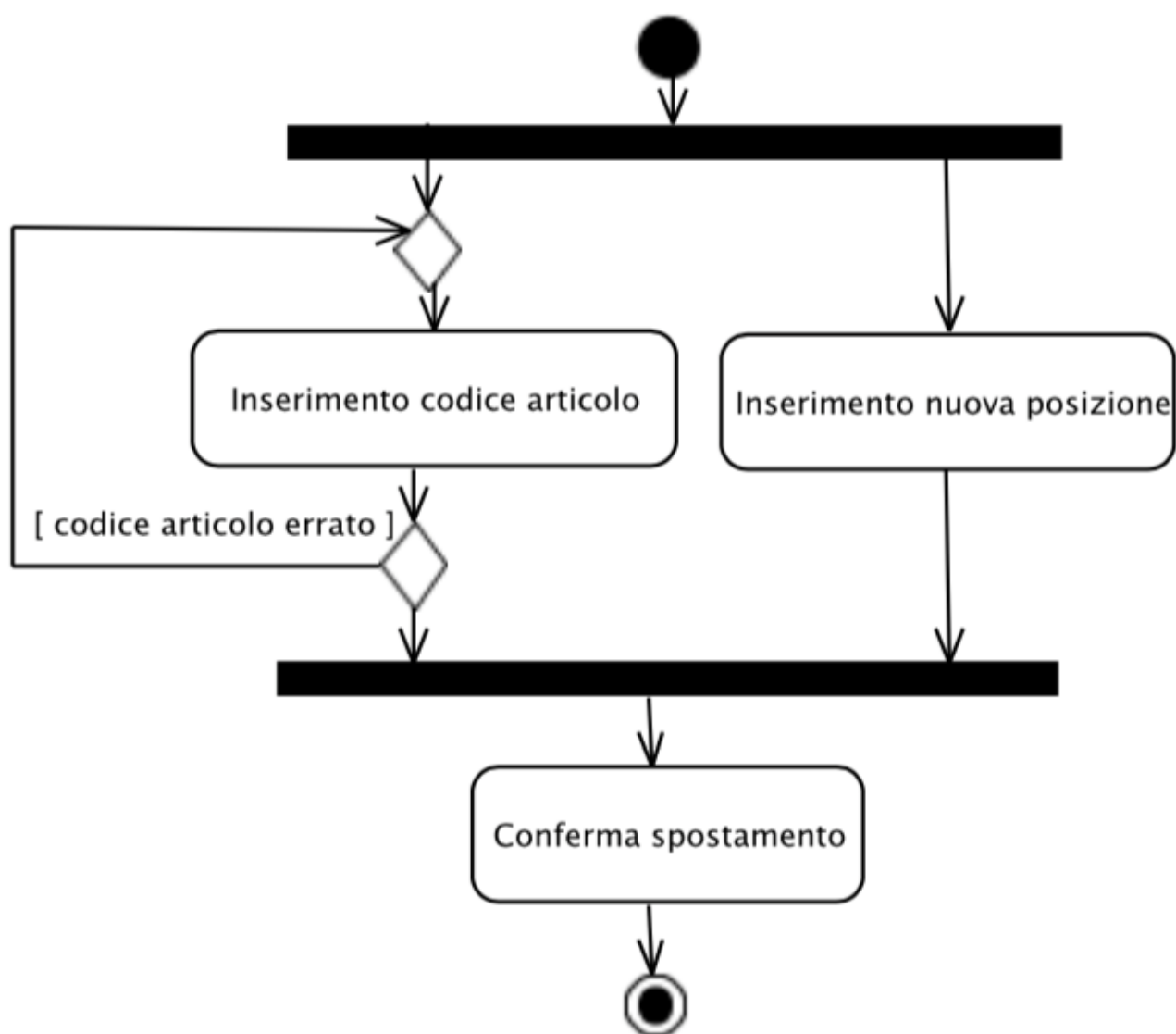
Visualizzare ordini passati:



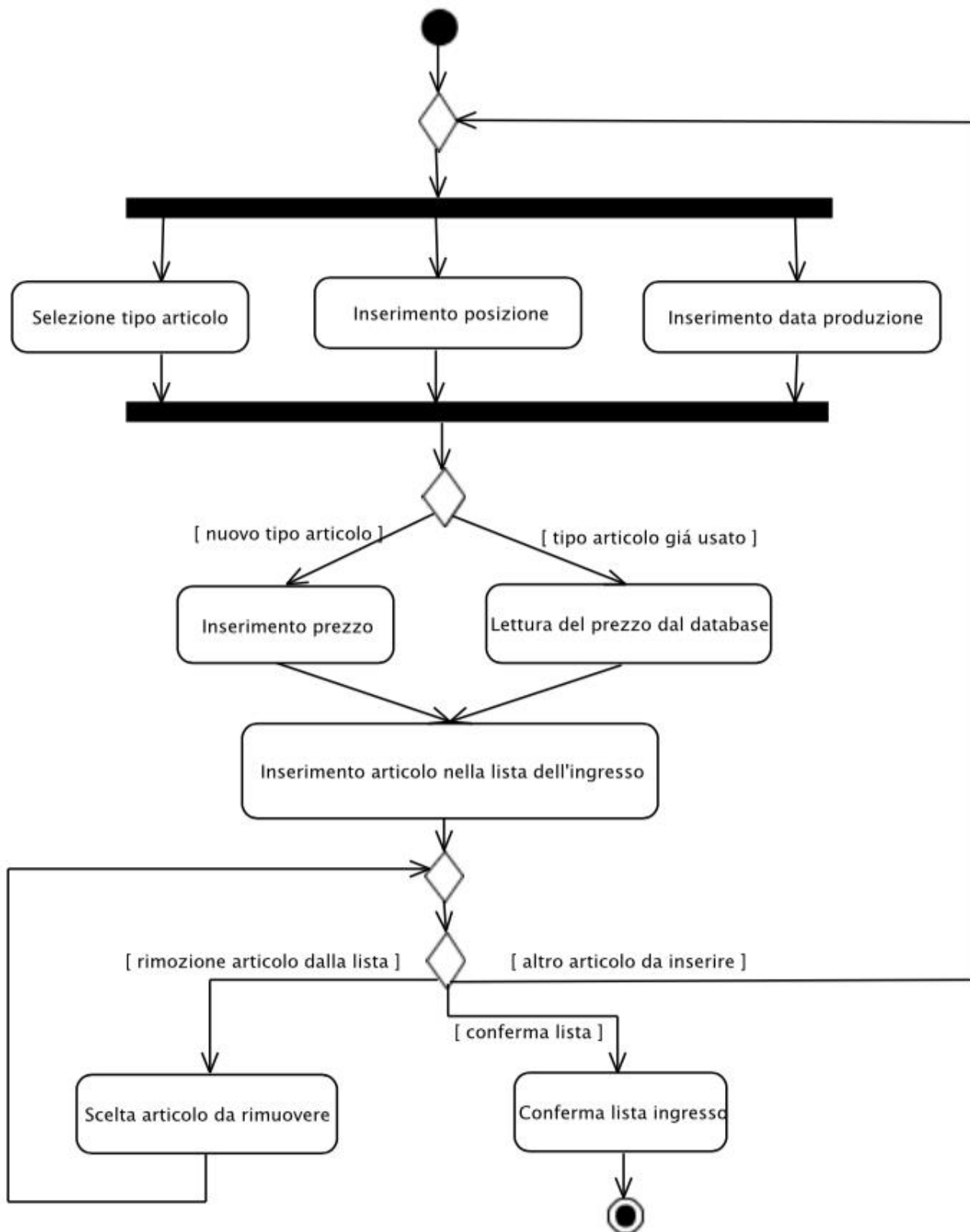
Effettuare un nuovo ordine:



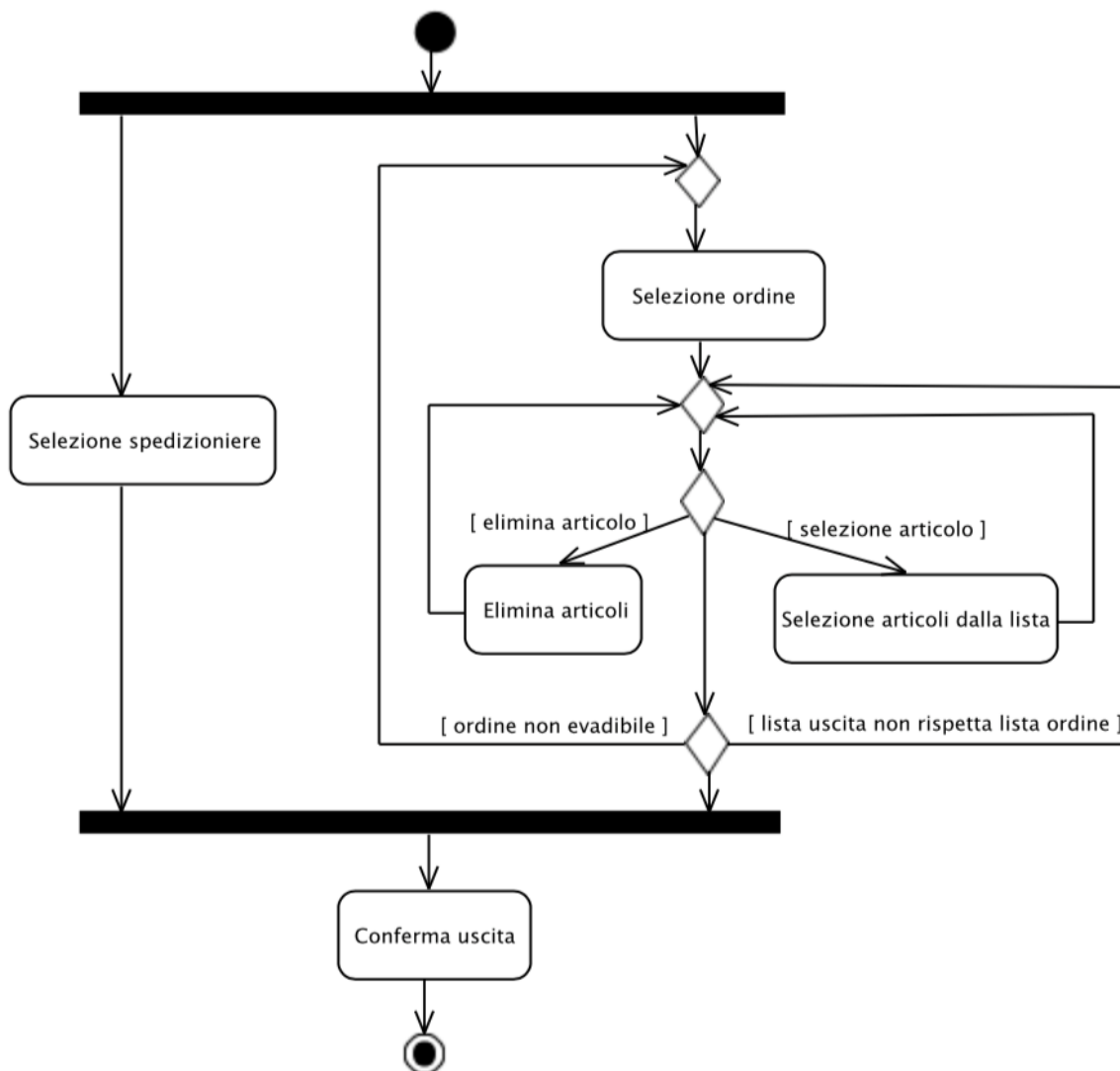
Spostamento di un articolo:



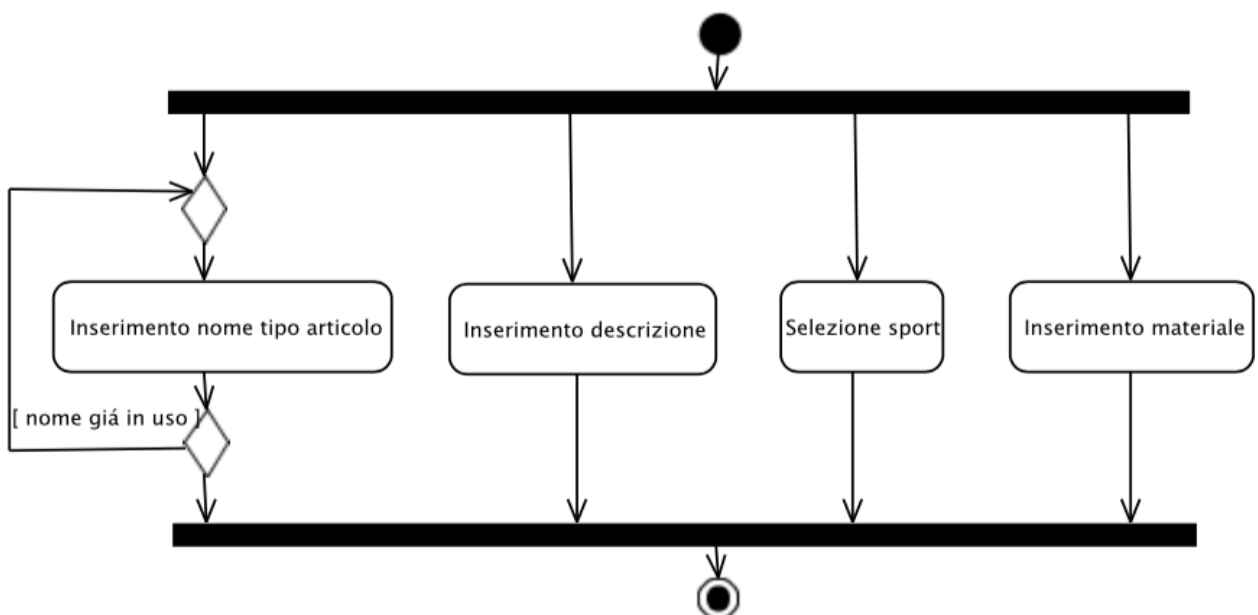
Registrazione di un nuovo ingresso:



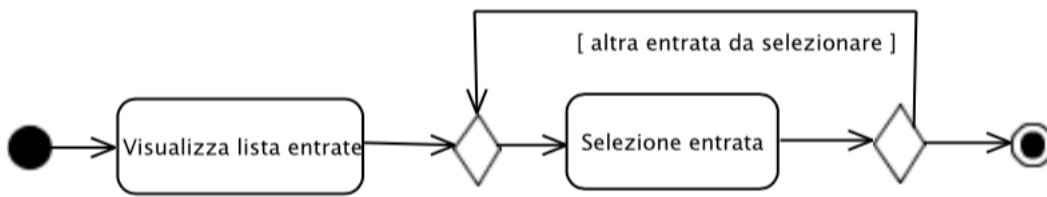
Registrazione di un'uscita:



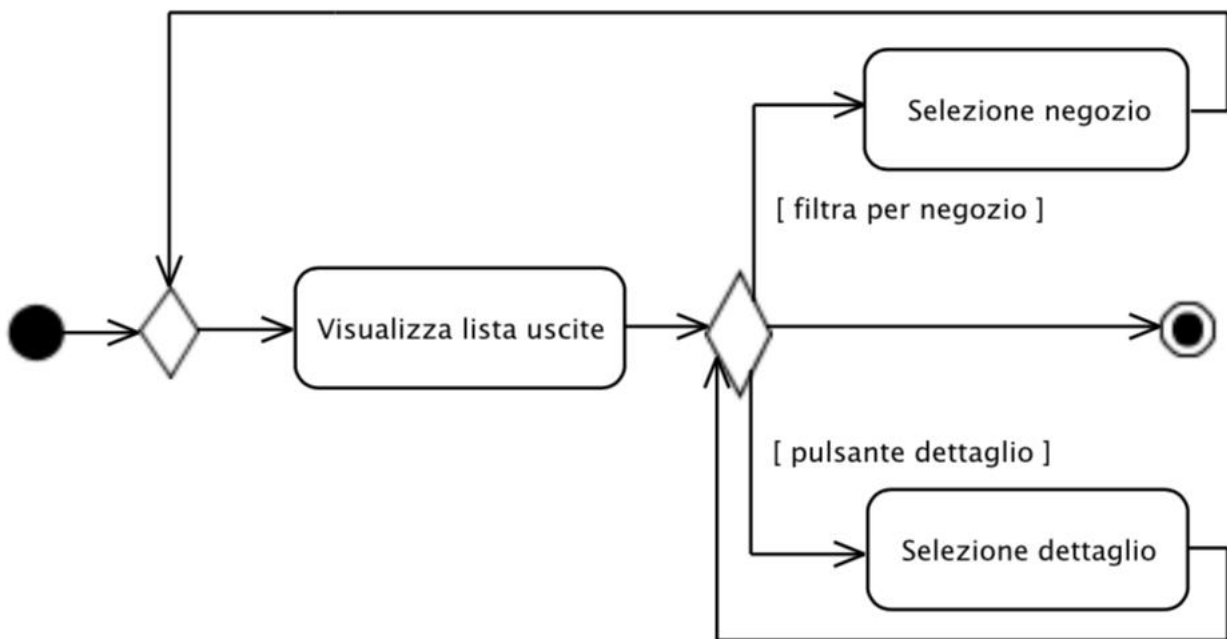
Inserimento nuovo tipo articolo:



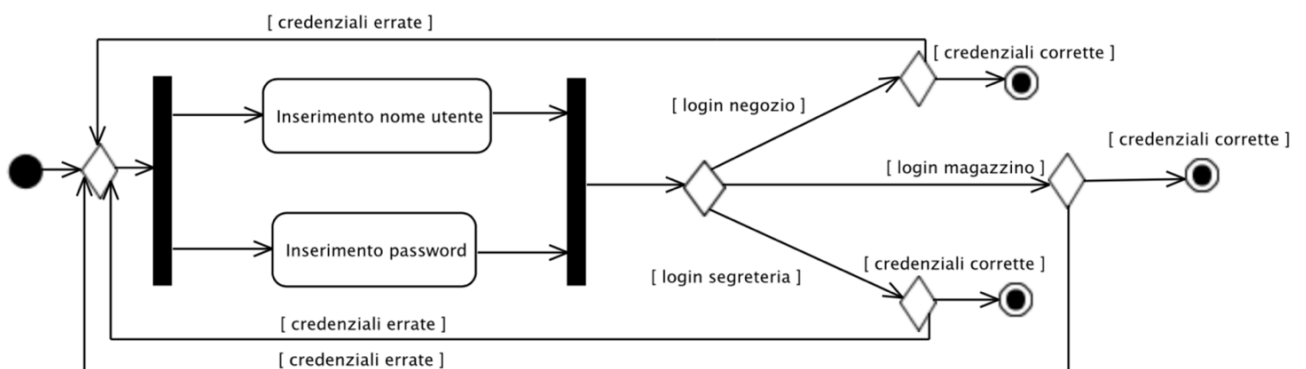
Visualizzazione lista delle entrate:



Visualizzazione lista delle uscite:



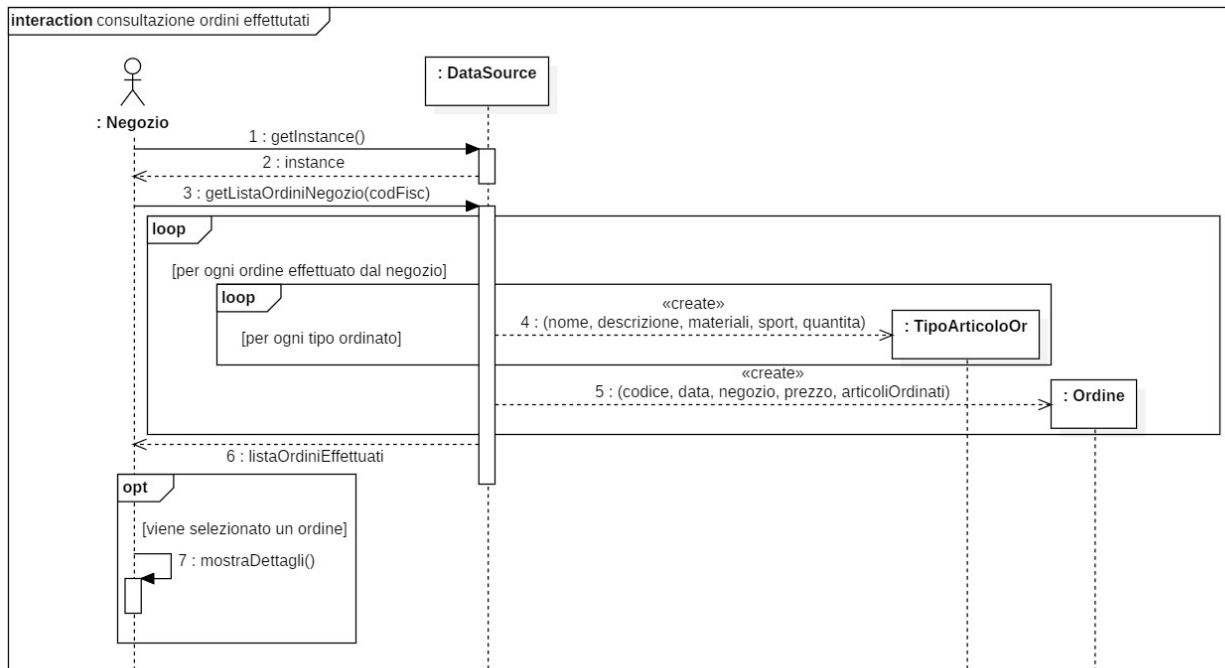
Effettuare il login:



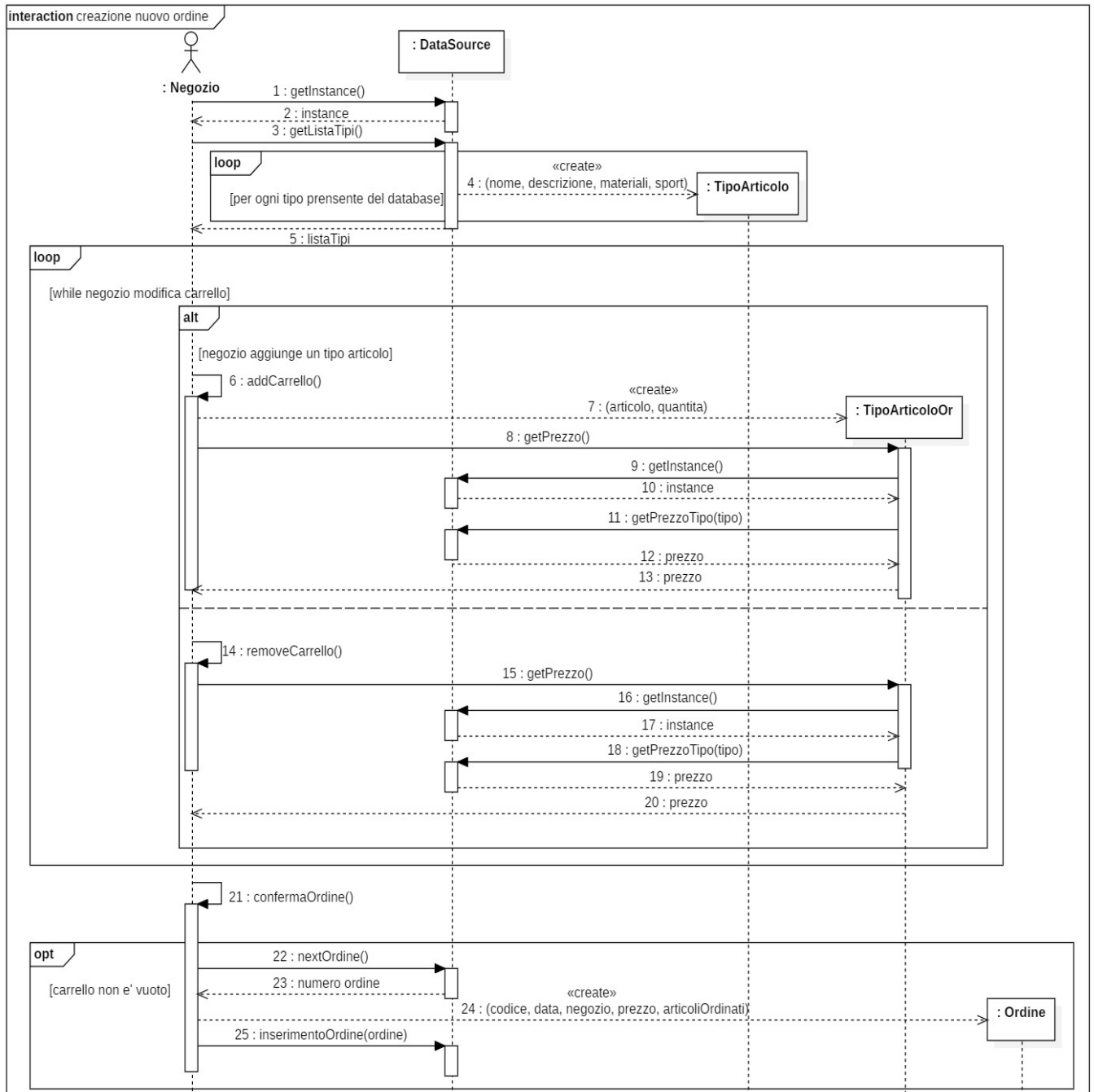
3.3 Diagrammi delle sequenze

Di seguito vengono presentati i diagrammi di sequenza che mostrano il flusso temporale di esecuzione dei possibili scenari che il prototipo presenta.

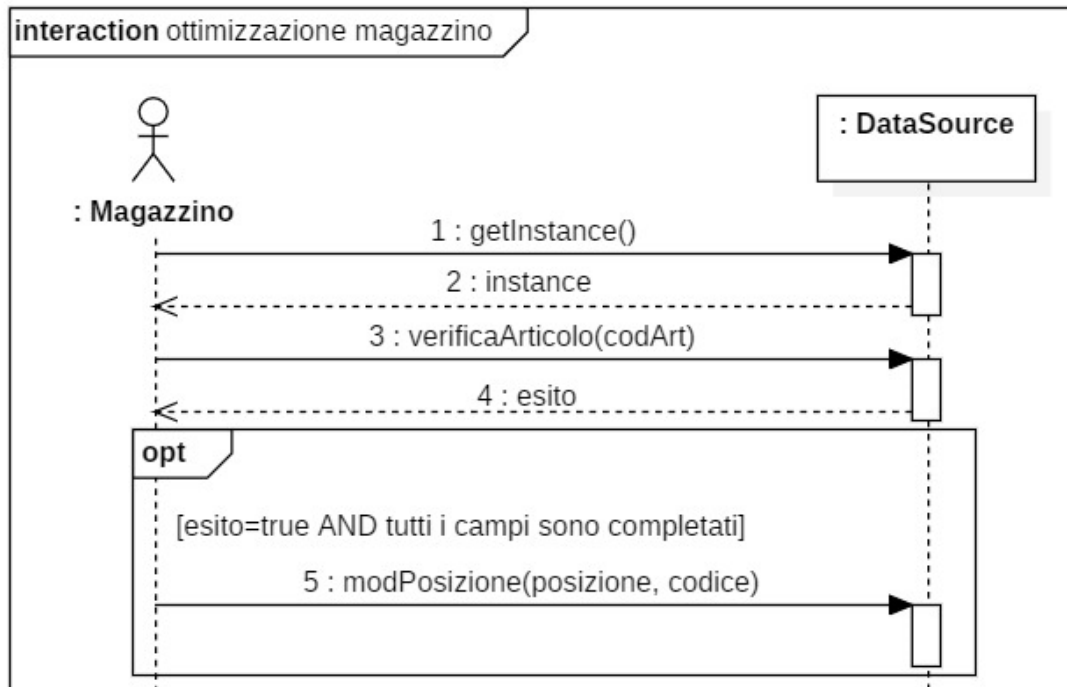
Consultazione ordini effettuati:



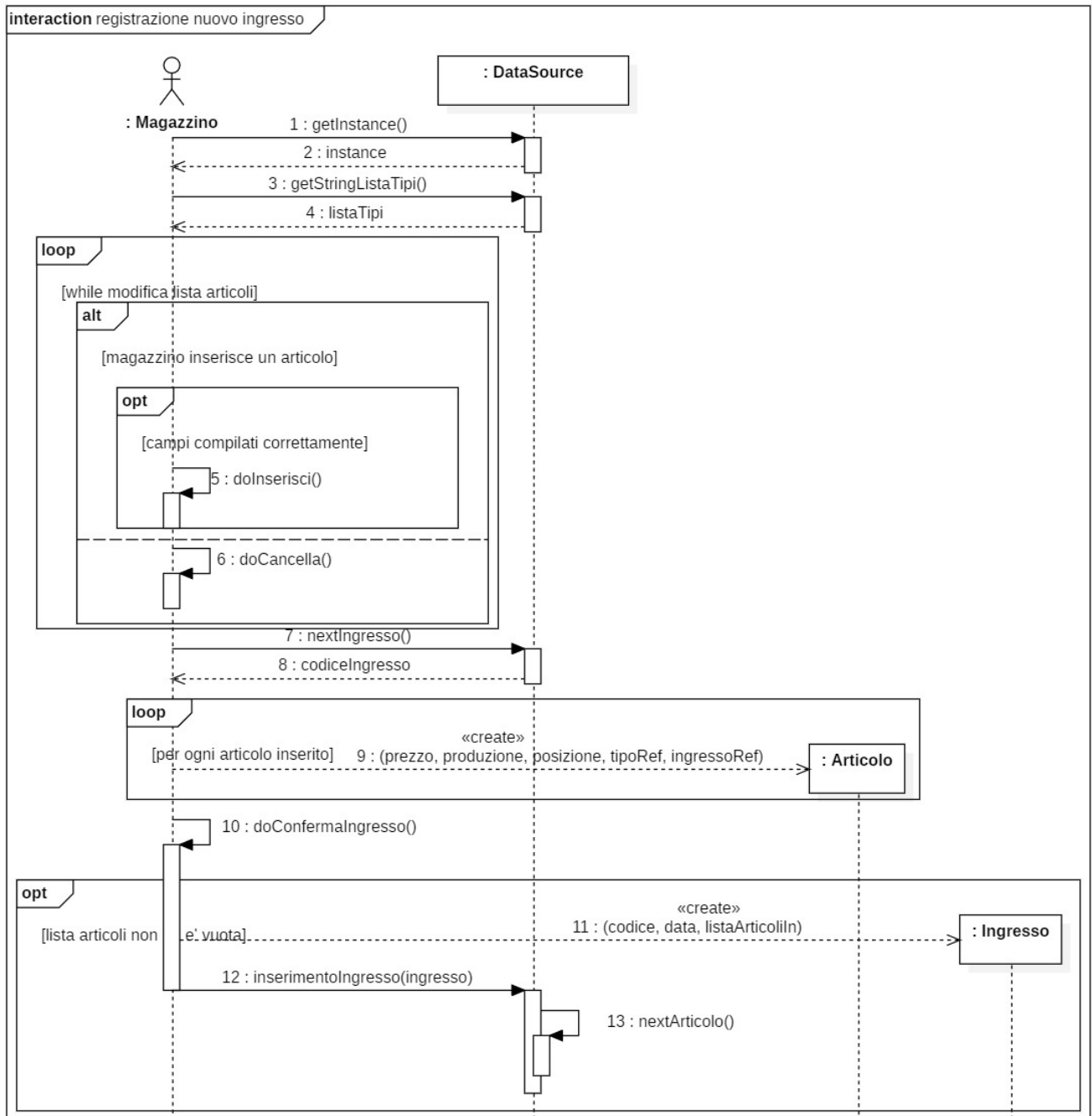
Creazione nuovo ordine:



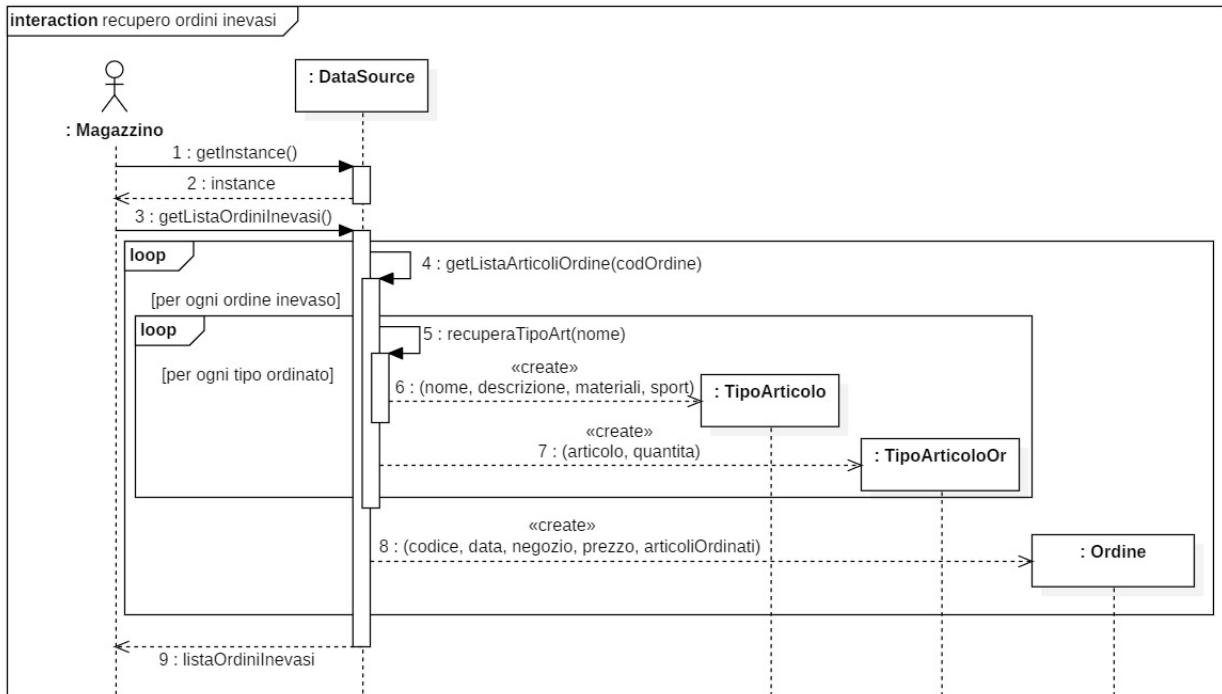
Spostamento di un articolo nel magazzino:



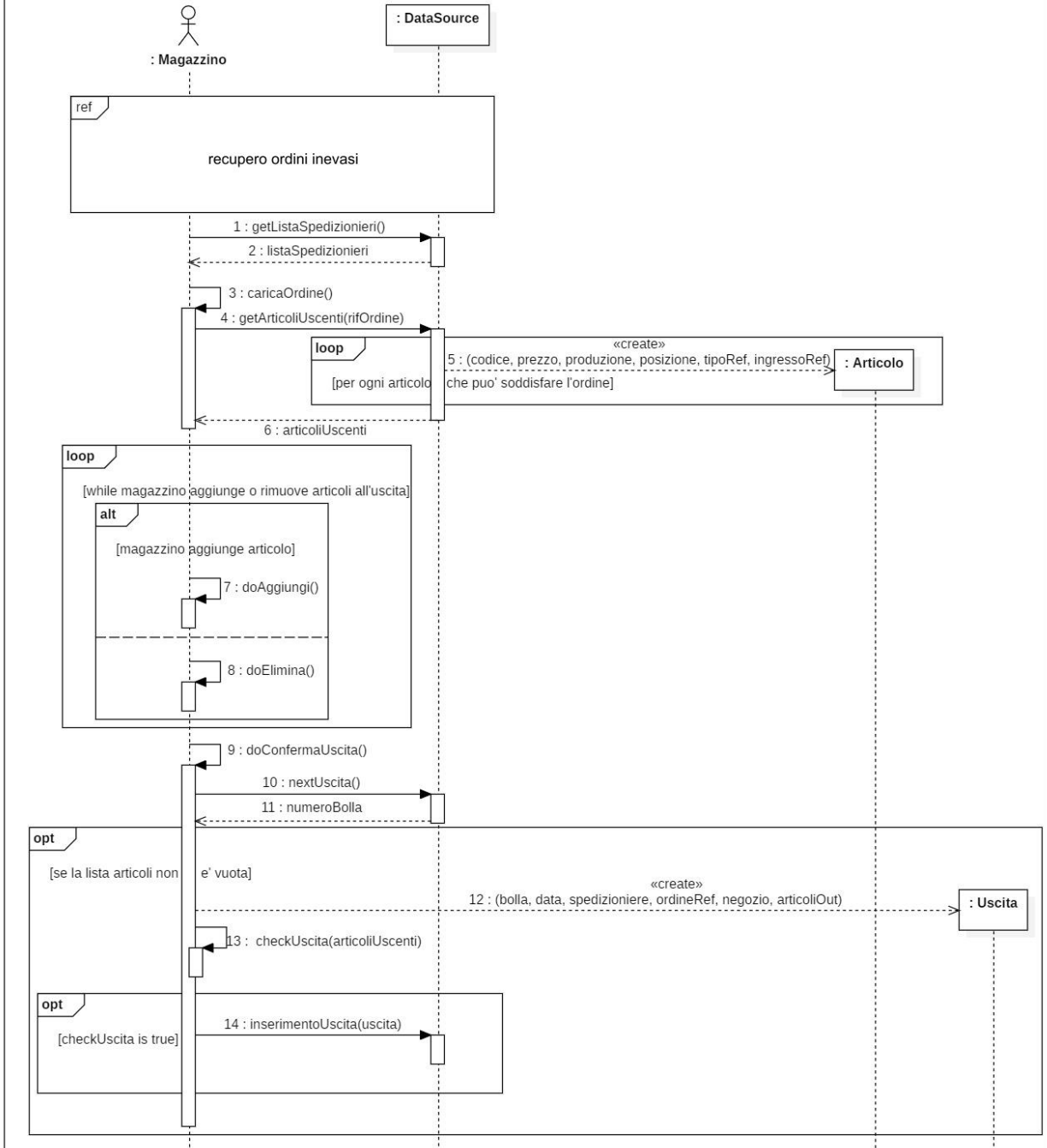
Registrazione di un ingresso:



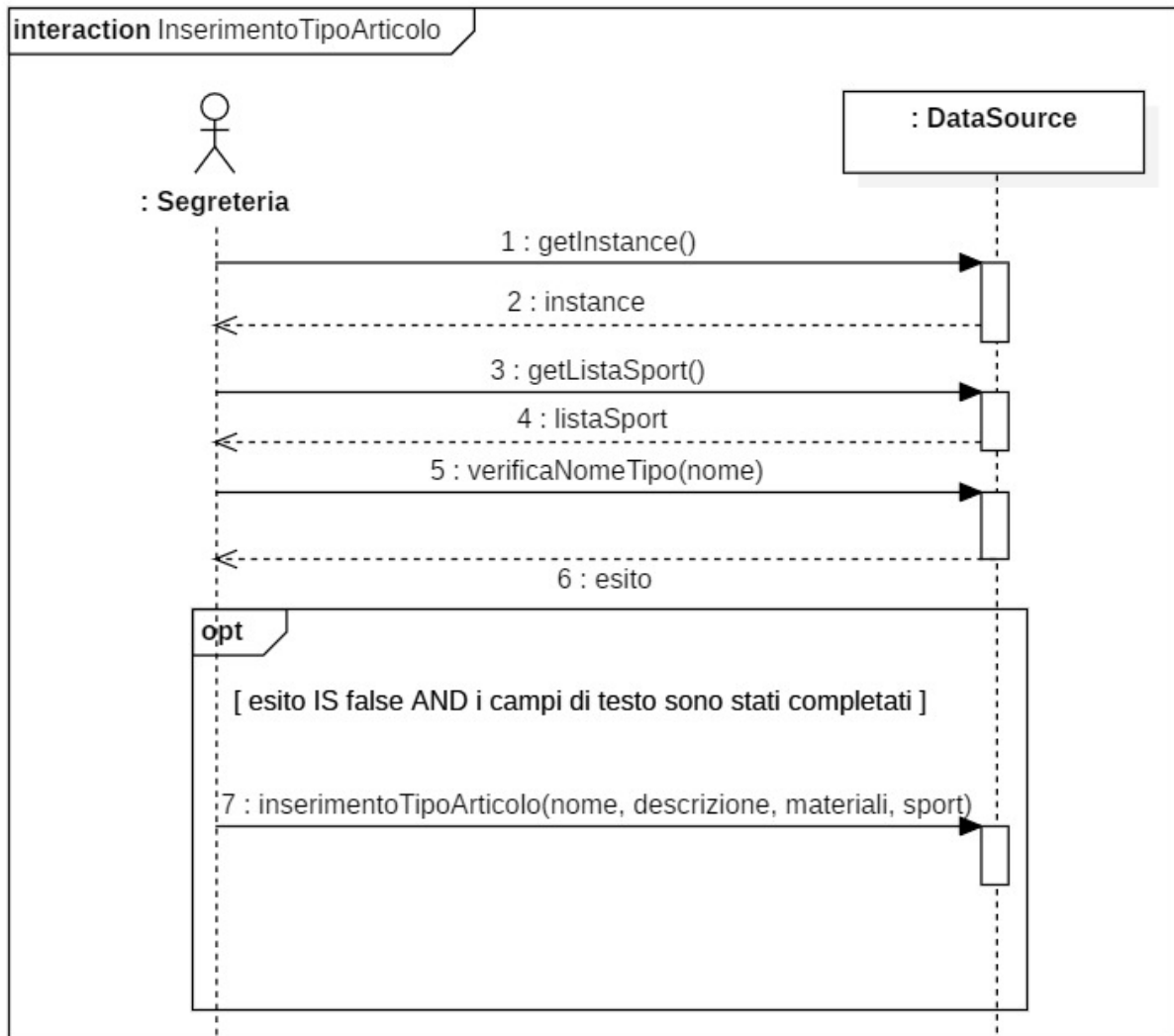
Registrazione di un'uscita:



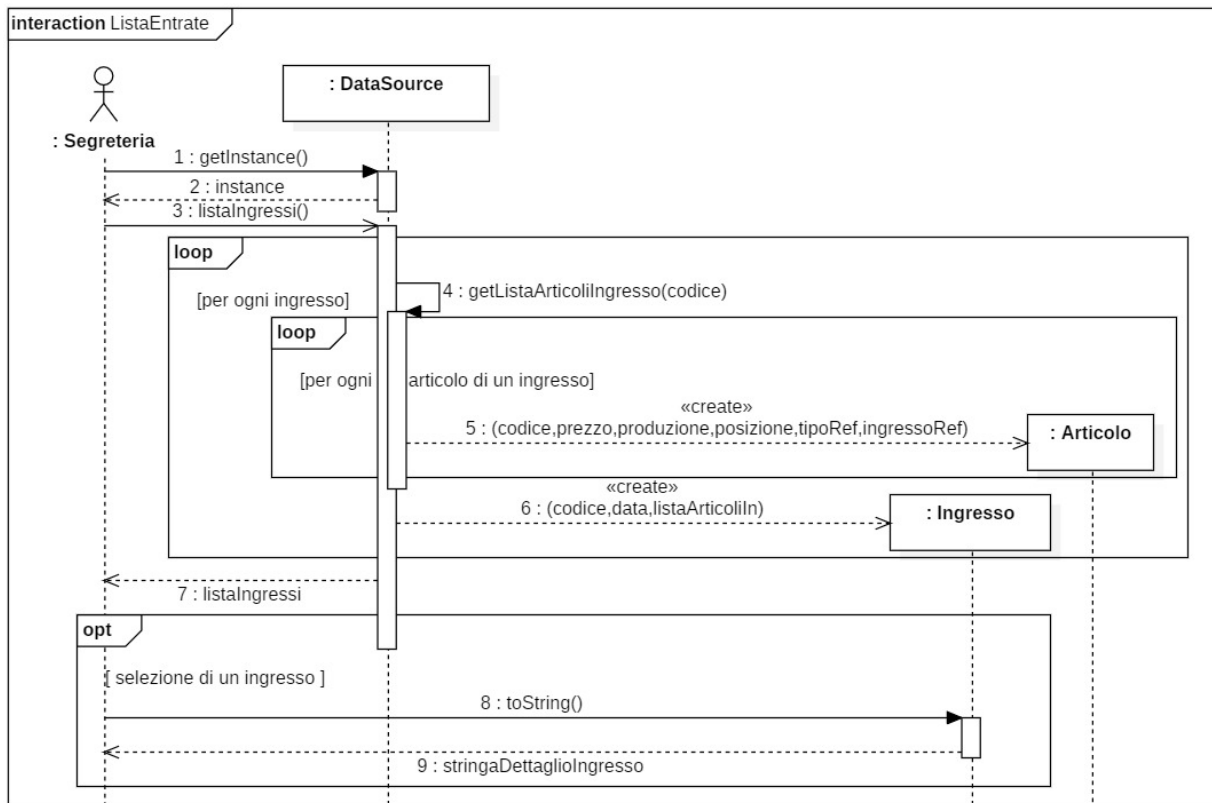
interaction registrazione nuova uscita



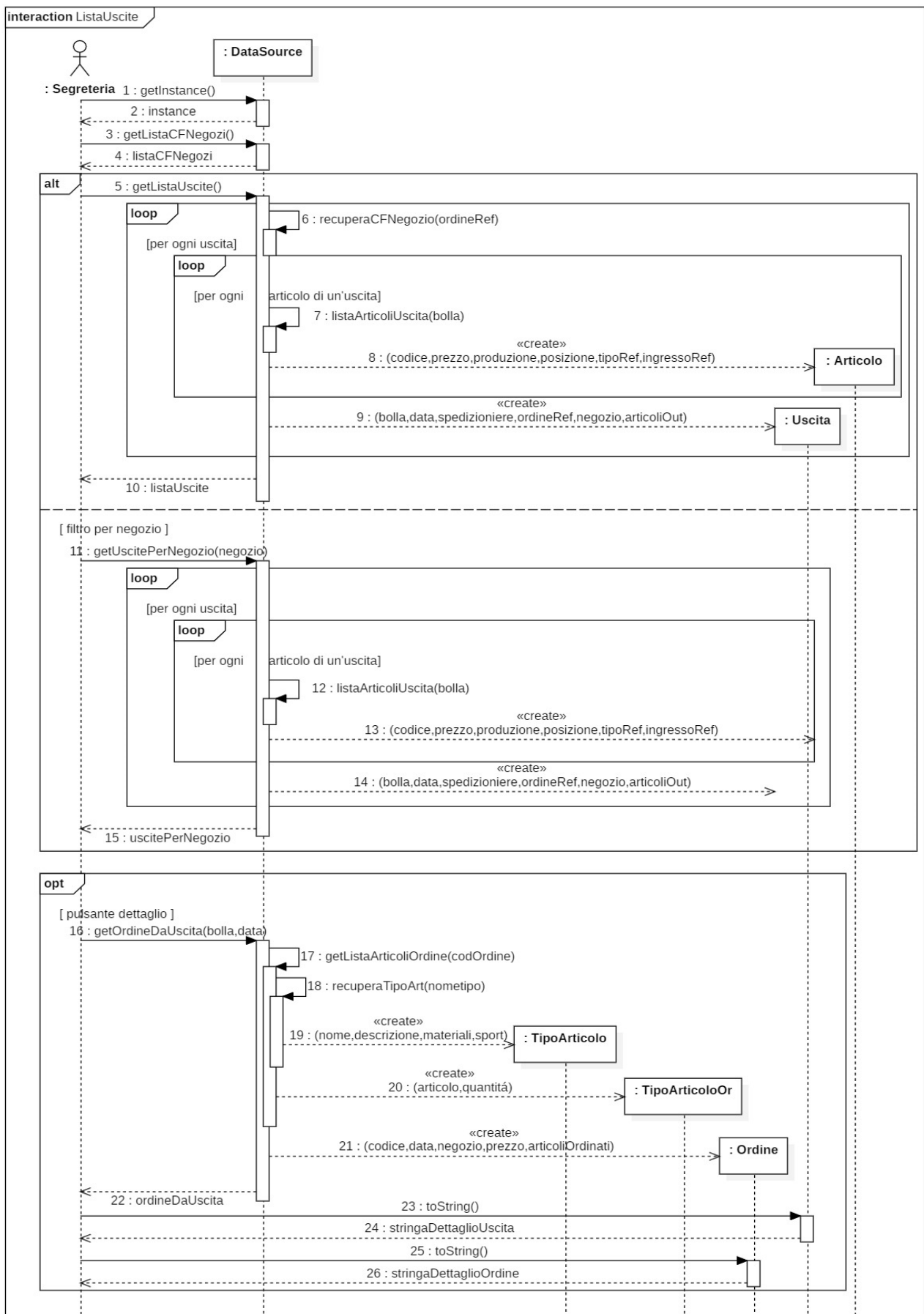
Inserimento nuovo tipo articolo:



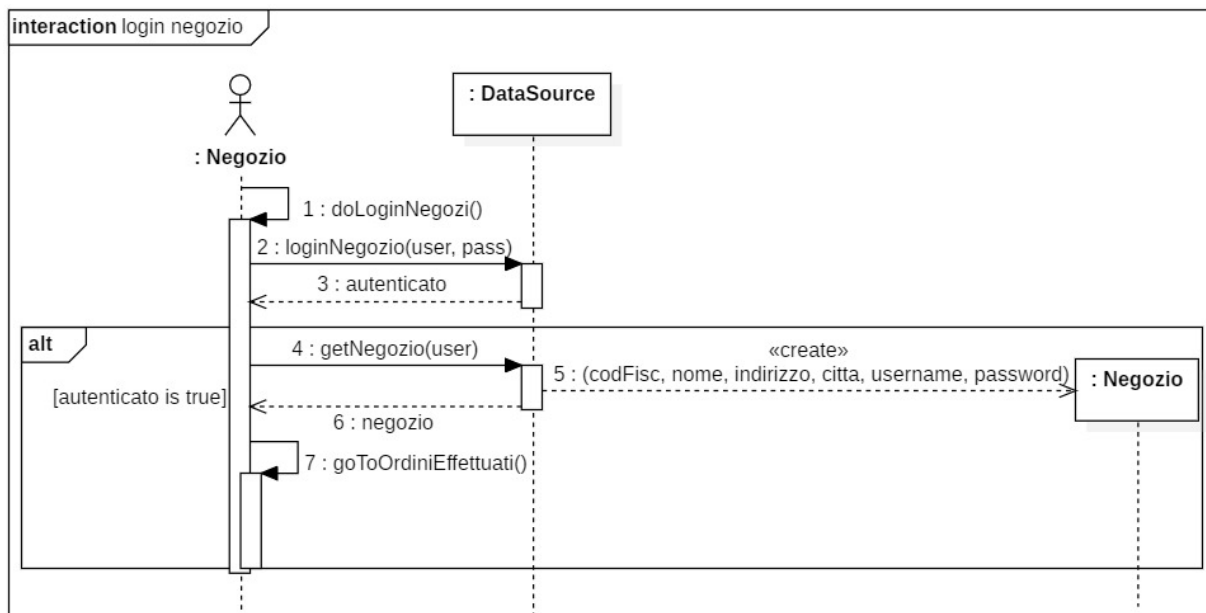
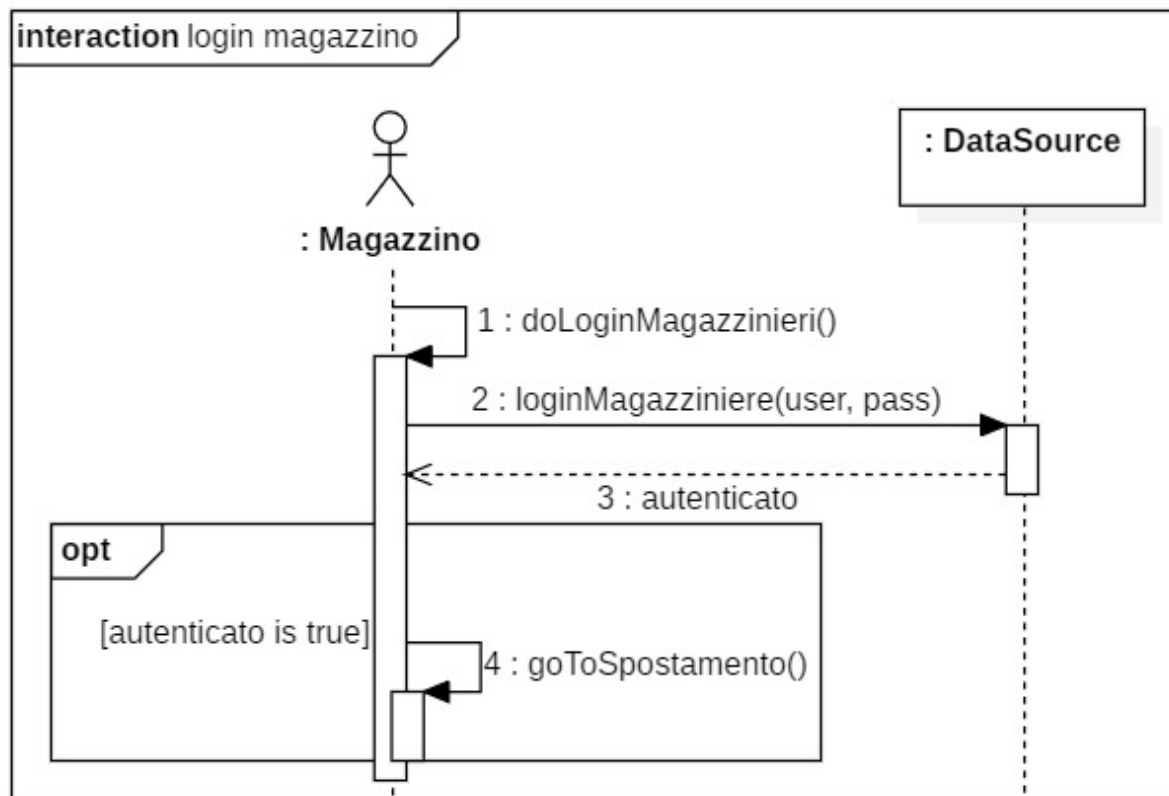
Visualizzazione lista delle entrate:

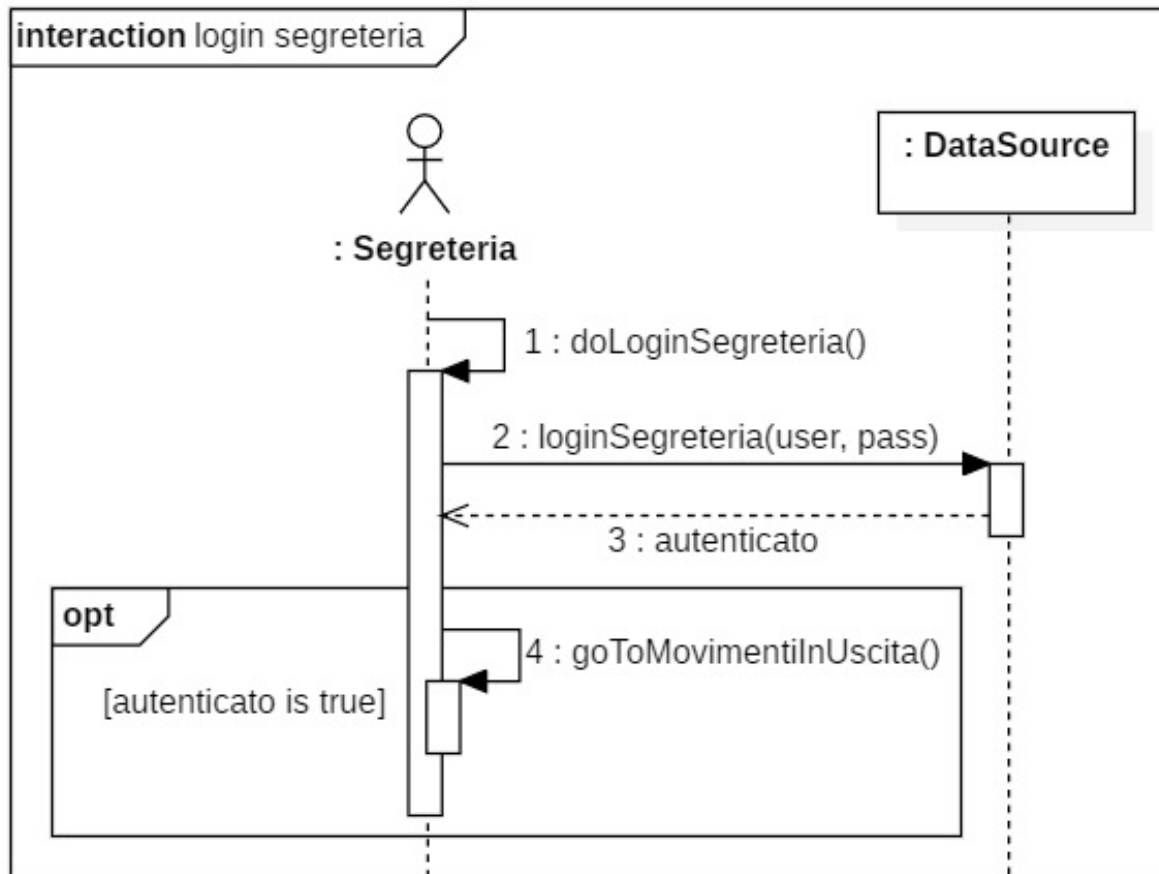


Visualizzazione lista delle uscite:

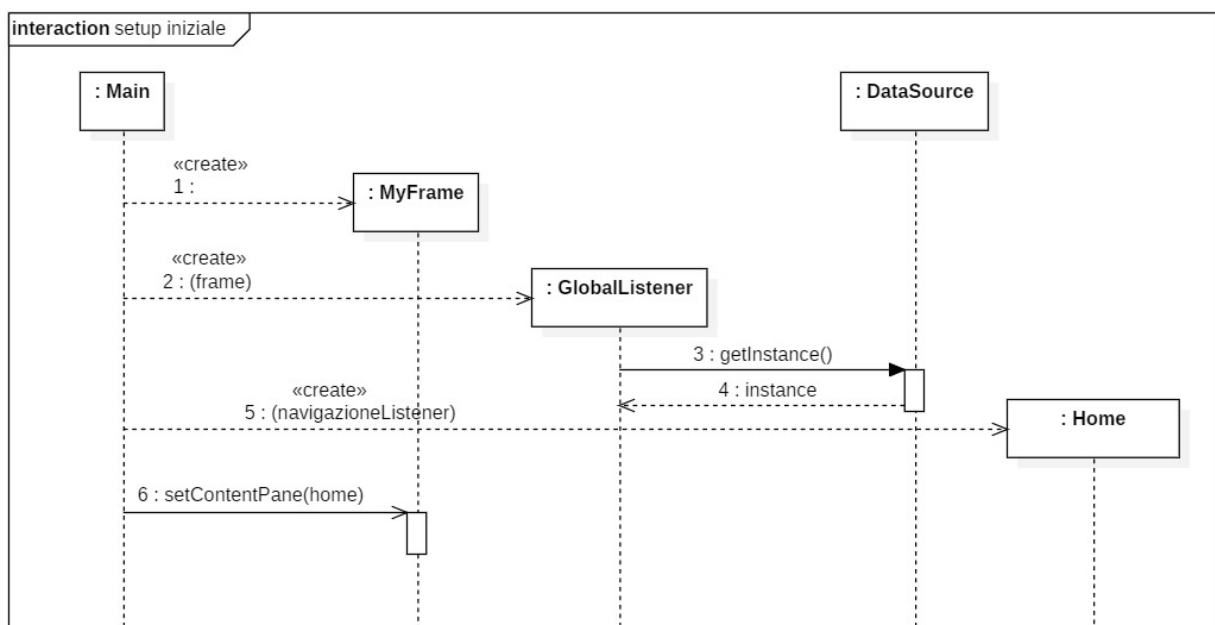


Login:

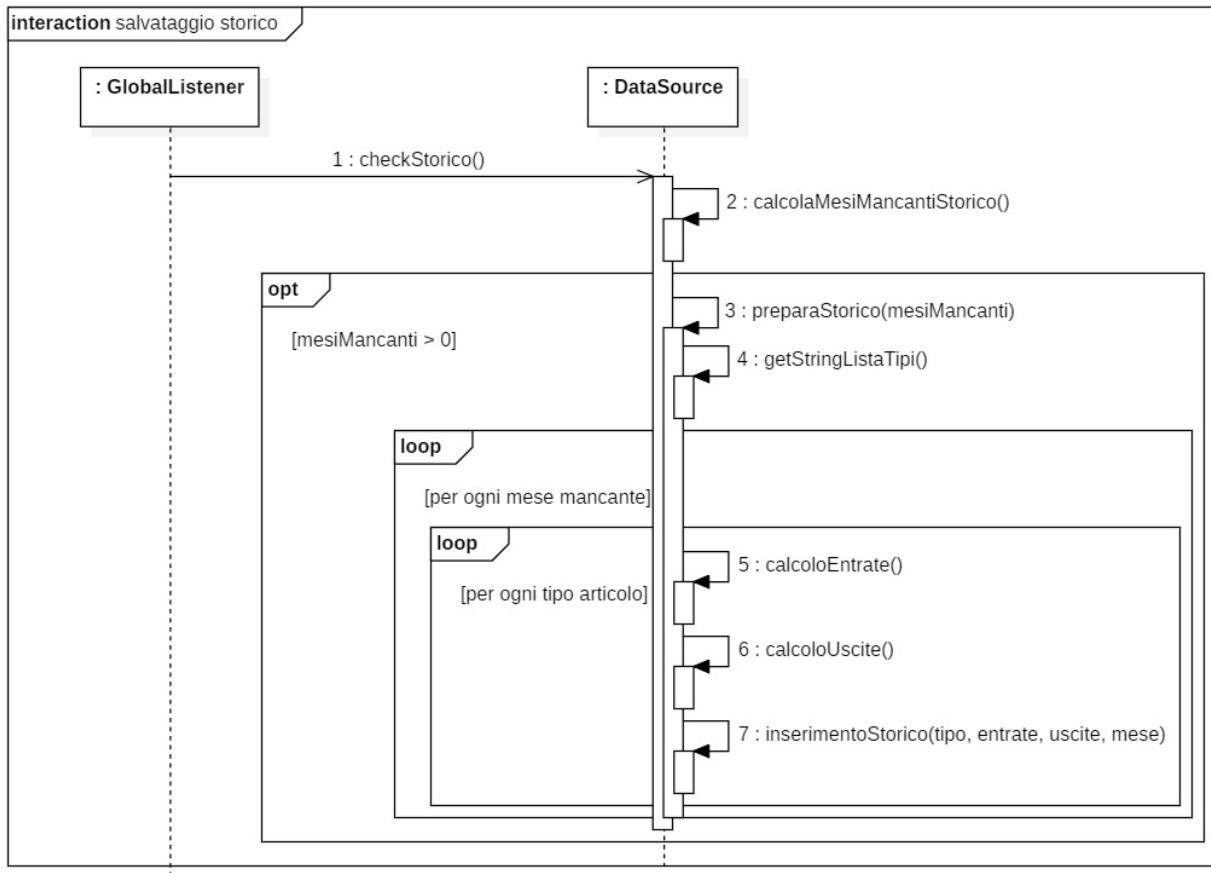




Setup iniziale del sistema:

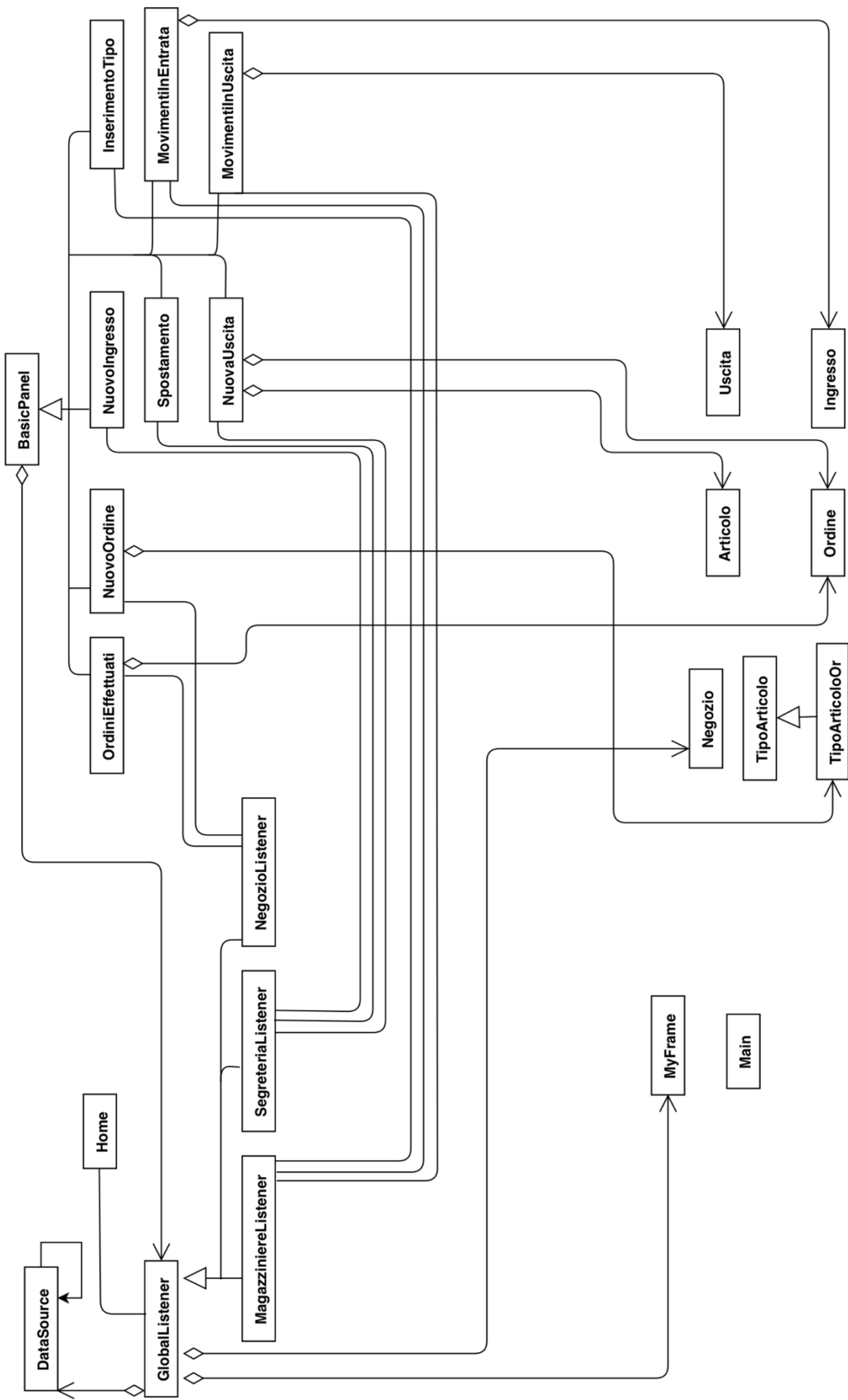


Aggiornamento periodico dello storico:

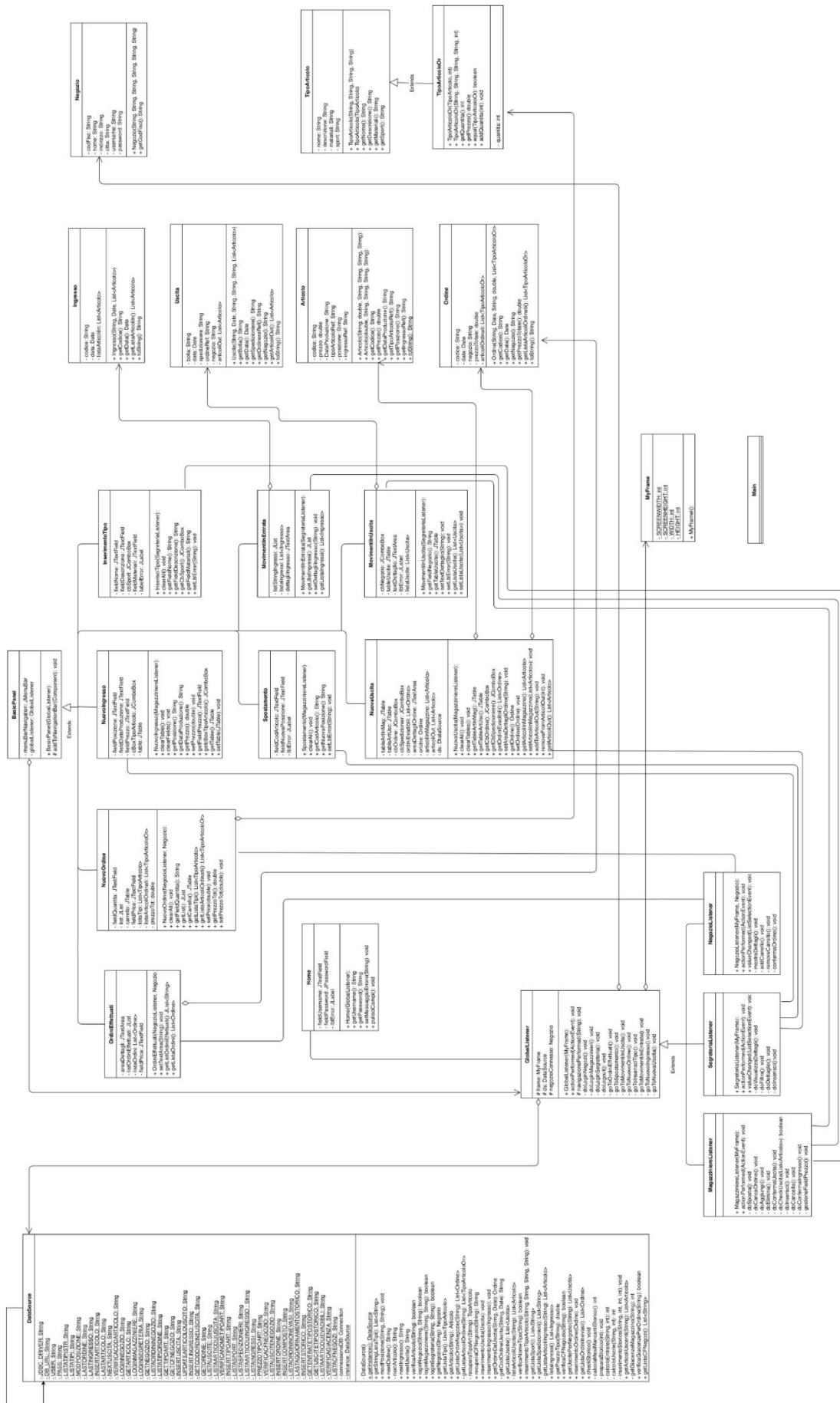


3.3 Diagramma delle classi

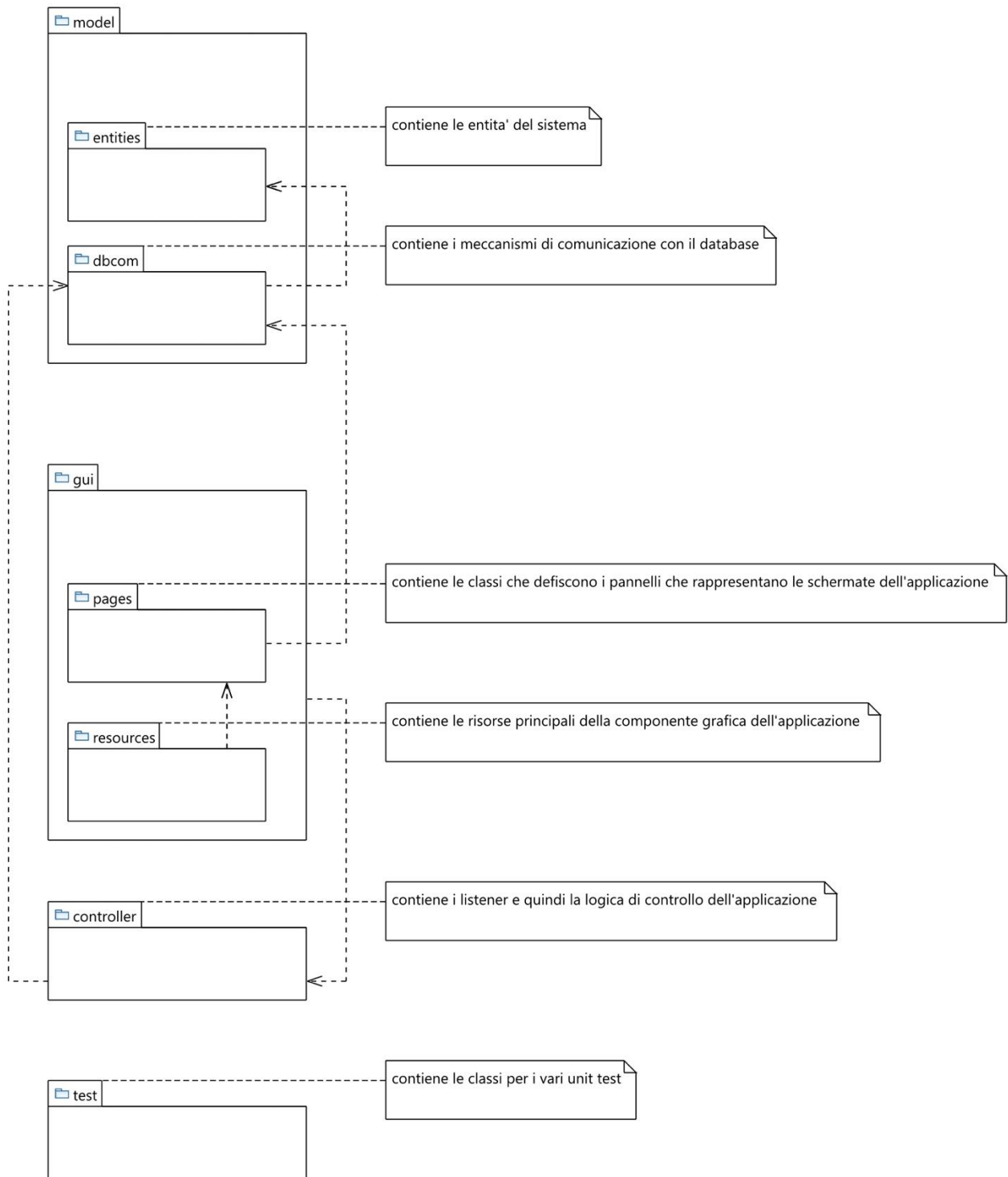
In versione semplificata:



In versione estesa:



3.4 Diagramma dei package



4 Scelte progettuali

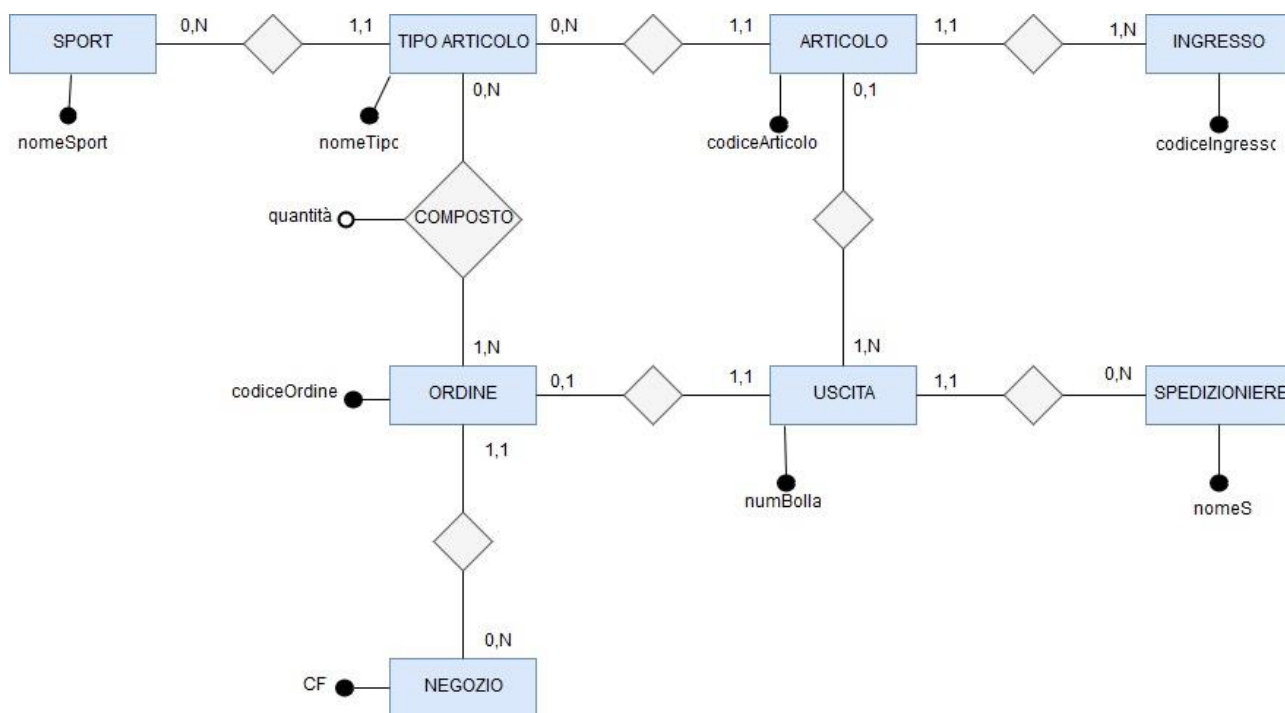
Abbiamo scelto di sviluppare il prototipo usando il linguaggio Java in quanto l'abbiamo ritenuto il linguaggio ideale per questo elaborato e in quanto faceva già parte del nostro bagaglio di conoscenze pregresse.

4.1 Database

Visto che avevamo necessità di un supporto in grado di mantenere i nostri dati in modo persistente, abbiamo scelto di affidarci a PostgreSQL. Lo abbiamo scelto in quanto lo avevamo studiato nel corso di Basi di dati e ci sembrava adatto alla realizzazione del progetto. È stata realizzata una classe di interfacciamento che provvede a nascondere la complessità delle transazioni (vedasi Facade Pattern).

4.1.1 Modellazione

Di seguito riportiamo il nostro modello:



Abbiamo scelto di non effettuare nessun controllo sui campi `DataProduzione` e `Posizione` in quanto la data di produzione non sempre è presente per un articolo e non è detto che sia completa di giorno e mese e lasciamo poi ai magazzinieri la scelta del formato in cui viene scritta la posizione. Inoltre data produzione la abbiamo considerata come una stringa visto che su di essa non viene effettuata nessuna operazione.

Per quanto riguarda l'aggiornamento dello storico viene effettuato un controllo ad ogni login per verificare che lo storico sia aggiornato e in caso contrario viene eseguita una funzione di aggiornamento dello stesso. Idealmente la funzione di aggiornamento viene svolta soltanto al momento del primo login di un mese.

Il prezzo lo abbiamo inserito all'interno della tabella articolo secondo quanto specificato nei requisiti.

Abbiamo fatto uso inoltre di tre tabelle separate dal core del nostro modello e che quindi figurano come tabelle a sé stanti all'interno del nostro database. Per quanto riguarda la tabella storico l'abbiamo separata così da mantenere la correttezza storica anche dei tipi di articolo in caso di rimozione di questi dalla tabella TipoArticolo, a discapito dell'aggiornamento nel caso di modifica del nome. Le tabelle magazzinoiere e segreteria invece sono a parte in quanto non sono coinvolte in nessuna relazione con il resto del nostro modello.

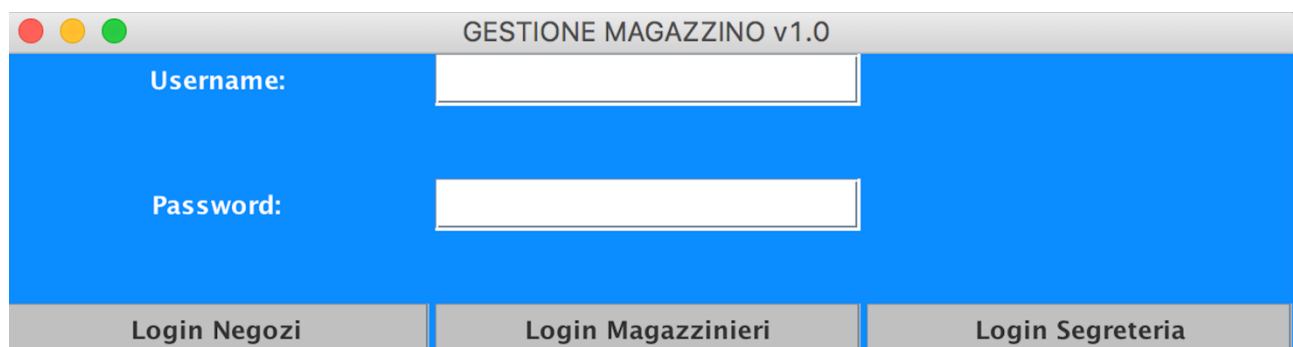
Abbiamo scelto inoltre di fare uso di sei indici che vengono usati nelle query chiamate più frequentemente, anche se con un database così ridotto i benefici non sono poi molti.

4.2 GUI

Per la realizzazione dell'interfaccia grafica ci siamo affidati alla libreria Swing in quanto ben si prestava al nostro scopo, per la sua realizzazione ci siamo affidati all'add-on "Swing Designer" per l'IDE Eclipse. Il risultato è un compromesso tra resa visiva finale e complessità dell'implementazione.

Di seguito vengono mostrate alcune significative schermate di esempio:

(schermata di login)



The screenshot shows a Java Swing window titled "GESTIONE MAGAZZINO v1.0". The window has a blue background and contains two text input fields. The first field is labeled "Username:" and the second is labeled "Password:". Below the input fields, there are three buttons: "Login Negozi", "Login Magazzinieri", and "Login Segreteria". The "Login Magazzinieri" button is currently selected, indicated by a blue highlight.

(inserimento nuovo ingresso)

GESTIONE MAGAZZINO v1.0

Benvenuto! Logout

Spostamento **Nuovo Ingresso** Nuova Uscita

Tipo Articolo: pallone in cuoio ▼

Posizione:

Data produzione:

Prezzo: 20.00

Inserisci

Riepilogo:

Tipo	Posizione	Data produzione	Prezzo
------	-----------	-----------------	--------

Cancella Conferma Ingresso

(visualizzazione lista delle uscite)

GESTIONE MAGAZZINO v1.0

Benvenuto! Logout

Uscite Entrate Inserimento Tipo Articolo

Negozio: Visualizza tutti ▼

Filtra

Movimenti In Uscita

Data	N. bolla
2018-03-18	1
2018-03-21	2
2018-03-26	3
2018-04-05	4
2018-04-20	5
2018-04-21	6
2018-05-27	7
2018-07-09	8
2018-07-09	9
2018-07-09	10
2018-07-09	11
2018-07-09	12
2018-07-09	13

Dettagli Uscita:

Data: 2018-03-18
Numero Bolla: 1
Spedizioniere: gls
Riferimento Ordine: 2
Lista Articoli:
1
5

Dettagli Ordine:

Codice: 2
Data: 2018-03-16
Negozio: lzzmnl
Prezzo totale: 30.00
Tipi Articolo:
pallone in cuoio x 1
pallone leggero x 1

Dettaglio

4.3 MVC pattern

Il pattern architetturale MVC lo abbiamo scelto in quanto ci permette di realizzare una distinzione tra il modello dei dati e la logica applicativa dalla logica di presentazione.

4.4 Singleton pattern

Abbiamo scelto di usare il pattern creazionale Singleton in quanto garantisce che ci sia una sola istanza attiva della classe DataSource e fornisce un punto di accesso globale alla stessa, la quale è l'unica classe incaricata all'accesso al database. L'utilizzo di questo pattern nella classe DataSource ci ha consentito un risparmio notevole di tempo rispetto alla nostra iniziale implementazione, ovvero l'uso di soli metodi statici in cui dovevamo creare una nuova connessione per ogni metodo. Ad esempio nell'esecuzione dei test del datasource ha abbassato i tempi da circa 10.0s a 0.5s, con un miglioramento quindi di oltre il 95%.

4.5 Facade pattern

Il pattern facade lo abbiamo adottato nella classe DataSource. Ci ha permesso di "mascherare" per mezzo di una classe la complessità e prolissità delle operazioni di accesso, interrogazione ed inserimento che riguardano il database fornendo metodi di più semplice utilizzo.

5 Test e validazione

Per la fase di testing abbiamo usato principalmente JUnit per la parte di codice relativa alla comunicazione con il database e test manuali per l'interfaccia grafica.

I test con JUnit ci hanno consentito un risparmio notevole di tempo in quanto per ciascuna nuova funzionalità si poteva verificare subito se questa era stata implementata correttamente.

Per la parte di test manuali sono stati svolti svariati test al fine di verificare la correttezza dei possibili parametri inseriti e la correttezza del passaggio degli stessi al resto del programma.