AA 2016/2017

Alessandro Righi

Università degli studi di Verona

Progetto ingegneria del software

Negozio CD

# Specifica del progetto

Si vuole progettare un negozio online di CD/DVD musicali descritto dalla seguente specifica:

Si vuole progettare un sistema informativo per gestire le informazioni relative alla gestione di un negozio virtuale di CD e DVD musicali (vende solo via web).

Il negozio mette in vendita CD di diversi generi: jazz, rock, classica, latin, folk, world-music, e così via. Per ogni CD o DVD il sistema memorizza: un codice univoco, il titolo, i titoli di tutti i pezzi contenuti, eventuali fotografie della copertina, il prezzo, la data dalla quale è presente sul sito web del negozio, il musicista/band titolare, una descrizione, il genere del CD o DVD, i musicisti che vi suonano, con il dettaglio degli strumenti musicali usati. Per ogni musicista il sistema registra il nome d’arte, il genere principale, l’anno di nascita, se noto, gli strumenti che suona.

Sul sito web del negozio è illustrato il catalogo dei prodotti in vendita.

Cliccando sul nome del prodotto, appare una finestra con i dettagli del prodotto stesso. I clienti possono acquistare on-line selezionando gli oggetti da mettere in un “carrello della spesa” virtuale.

Deve essere possibile visualizzare il contenuto del carrello, modificare il contenuto del carrello, togliendo alcuni articoli.

Al termine dell’acquisto va gestito il pagamento, che può avvenire con diverse modalità.

Il sistema supporta differenti ricerche: per genere, per titolare del CD o DVD, per musicista partecipante, per prezzo. Coerentemente, differenti modalità di visualizzazione, sono altresì supportate.

Ogni vendita viene registrata indicando il cliente che ha acquistato, i prodotti acquistati, il prezzo complessivo, la data di acquisto, l’ora, l’indirizzo IP del PC da cui è stato effettuato l’acquisto, la modalità di pagamento (bonifico, carta di credito, paypal) e la modalità di consegna (corriere, posta, …).

Per ogni cliente il sistema registra: il suo codice fiscale, il nome utente (univoco) con cui si è registrato, la sua password, il nome, il cognome, la città di residenza, il numero di telefono ed eventualmente il numero di cellulare.

Per i clienti autenticati, il sistema propone pagine specializzate che mostrano suggerimenti basati sul genere dei precedenti prodotti acquistati.

Se il cliente ha fatto già 3 acquisti superiori ai 250 euro l’uno entro l’anno, il sistema gli propone sconti e consegna senza spese di spedizione.

Il personale autorizzato del negozio può inserire tutti i dati dei CD e DVD in vendita. Il personale inserisce anche il numero di pezzi a magazzino. Il sistema tiene aggiornato il numero dei pezzi a magazzino durante la vendita e avvisa il personale del negozio quando un articolo (CD o DVD) scende sotto i 2 pezzi presenti in magazzino.

# Analisi dei requisiti

Si evidenziano i seguenti requisiti funzionali:

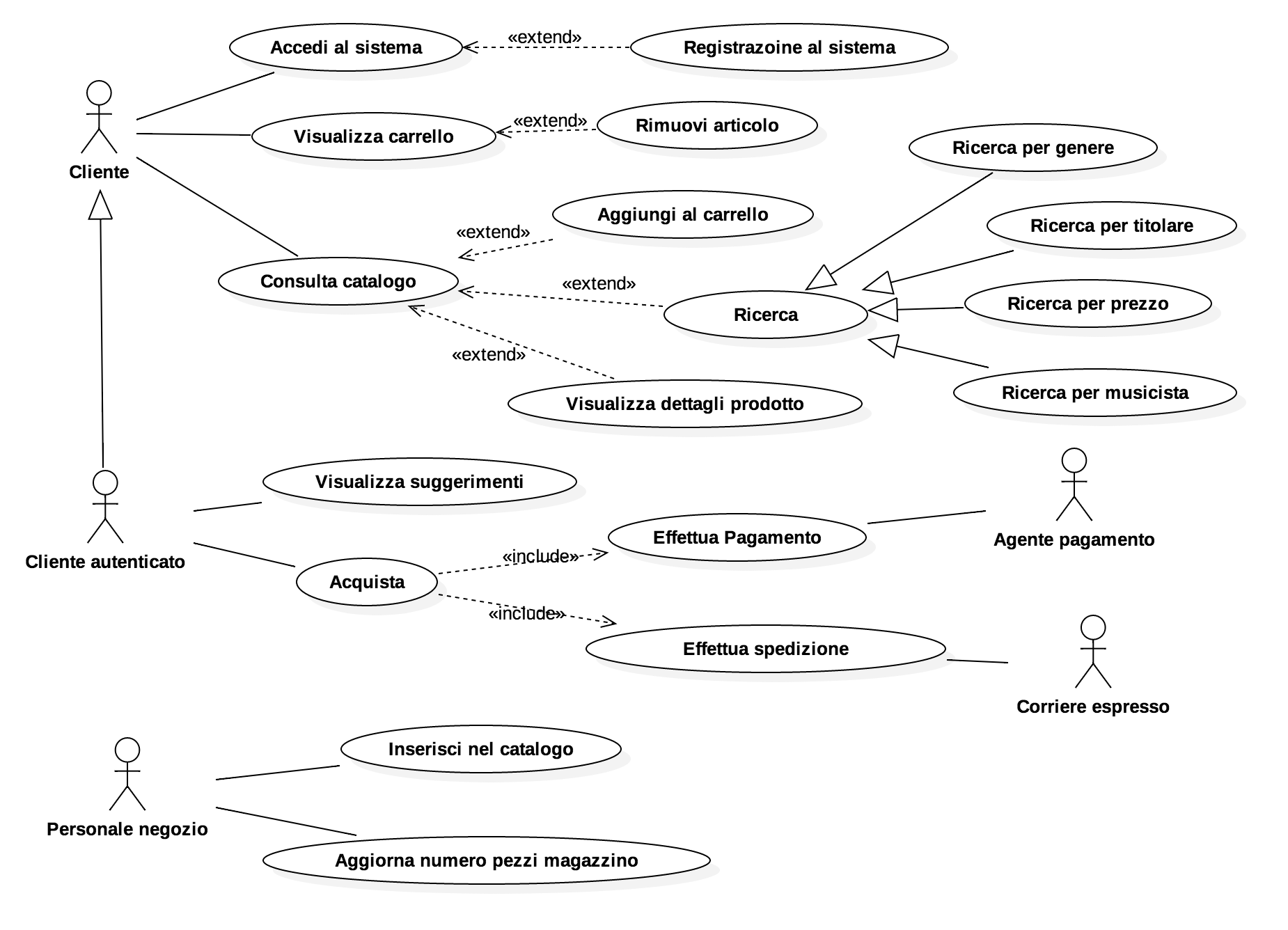
* Il negozio mette in vendita CD e DVD di diversi generi (rock, jazz, classica, latin, folk, world-music,...)
* Per ogni CD/DVD il sistema registra: codice univoco, titolo, titolo dei brani contenuti, foto di copertina, prezzo, data dalla quale è presente nel negozio, musicista/band titolare, descrizione, genere musicale, musicisti che suonano con dettaglio strumenti suonati
* Per ogni musicista il sistema registra: nome d’arte, genere principale, anno di nascita, strumenti che suona
* Sul sito web del negozio è visualizzato il catalogo dei prodotti in vendita
* Cliccando sul nome prodotto, appare una finestra con i dettagli del prodotto stesso
* I clienti possono acquistare online mettendo nel carrello della spesa
* Deve essere possibile visualizzare e modificare il contenuto del carrello
* Il pagamento può avvenire con diverse modalità: PayPal, bonifico, carta di credito
* Il sistema supporta ricerche per: genere, titolare del CD/DVD, musicista partecipante, prezzo.
* Per ogni vendita il sistema registra: il cliente che ha acquistato, i prodotti acquistati, il prezzo complessivo, la data di acquisto, l’ora, l’indirizzo IP del computer che ha effettuato l’acquisto, la modalità di pagamento (bonifico, carta di credito, paypal), le modalità di consegna (corriere, posta)
* Per ogni cliente il sistema registra: il suo codice fiscale, nome utente univoco, password, nome, cognome, città di residenza, numero di telefono ed eventualmente numero di cellulare.
* Per i clienti autenticati, il sistema propone pagine specializzate che mostrano suggerimenti basati sul genere dei precedenti prodotti acquistati.
* Se il cliente ha fatto 3 acquisti superiori a 250 euro l’uno entro l’anno, il sistema propone sconti e consegna senza spese di spedizione.
* Il personale autorizzato del negozio può inserire tutti i dati dei CD e DVD in vendita e il numero di pezzi a magazzino.
* Il sistema tiene aggiornato il numero dei pezzi a magazzino durante la vendita e avvisa il personale del negozio quando un articolo scende sotto i due pezzi presenti in magazzino.

Inoltre si evidenziano i seguenti requisiti non funzionali:

* Sicurezza
* Robustezza
* Prestazioni

# Analisi dei casi d’uso

## Use case diagram



## Attori principali

* **Cliente** indica un cliente che si connette al negozio senza autenticarsi
* **Cliente autenticato** indica un cliente registrato che ha effettuato l’autenticazione al negozio
* **Personale negozio** indica un dipendente del negozio che ha il compito di amministrare il sistema
* **Corriere espresso** indica il corriere che il sistema contatta per effettuare la spedizione
* **Agente pagamento** indica l’agente che si occupa di gestire il pagamento

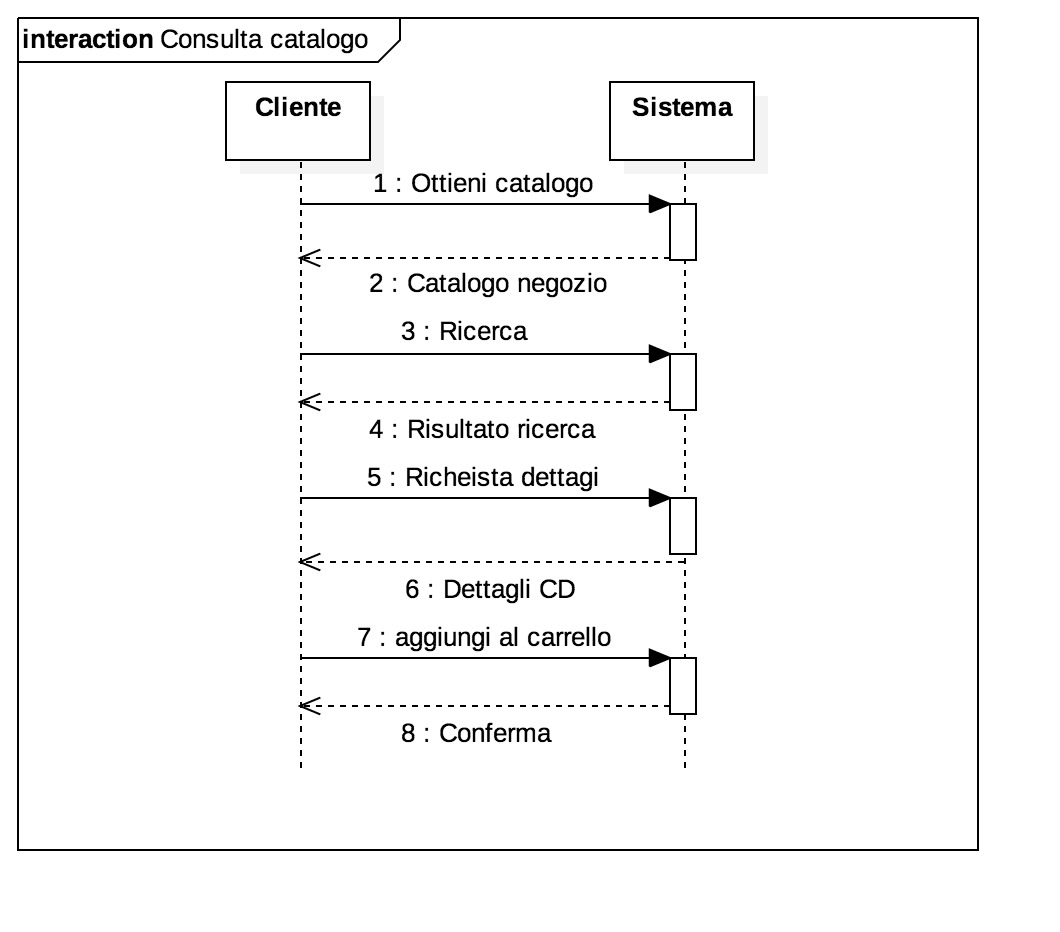
# Specifica casi d’uso principali

## Consulta catalogo

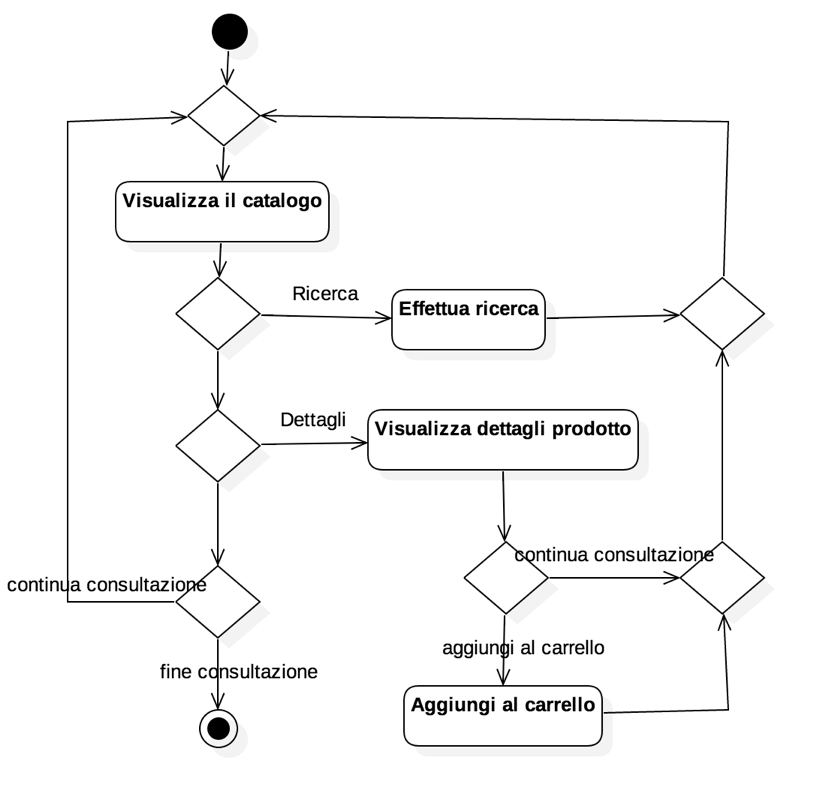
### Scheda di specifica

|  |  |
| --- | --- |
| ****Caso d’uso**** | **Consulta catalogo** |
| ID | UC1 |
| Attori | Cliente |
| Precondizioni | 1. Il cliente è connesso al sito web del negozio |
| Postcondizioni | 1. Il cliente ha eventualmente inserito nel carrello gli oggetti che intende acquistare |
| Sequenza degli eventi | 1. Il caso d’uso inizia quando il cliente si connette al sito web del negozio 2. Il sistema mostra al cliente delle pagine in cui sono visualizzati i CD/DVD presenti nel catalogo del negozio 3. Se il cliente clicca su un CD/DVD    1. Il sistema mostra una finestra riportante i dettagli del prodotto selezionato.    2. Se il cliente clicca su aggiungi al carrello:       1. Il prodotto selezionato viene aggiunto al carrello 4. Se il cliente inserisce dei parametri di ricerca e clicca su cerca:    1. Il sistema mostra solo i CD/DVD che soddisfano tali parametri |

Sequence diagram



### Activity diagram

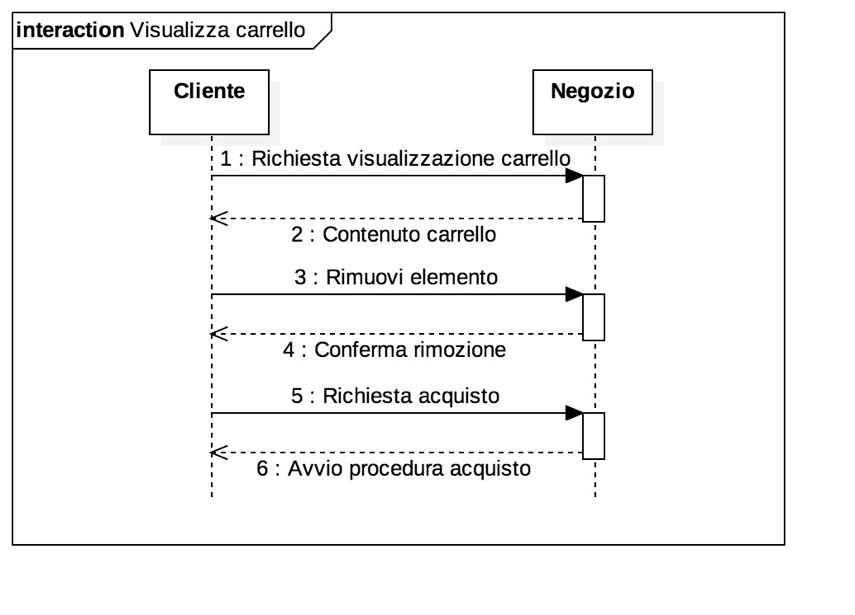


## Visualizza Carrello

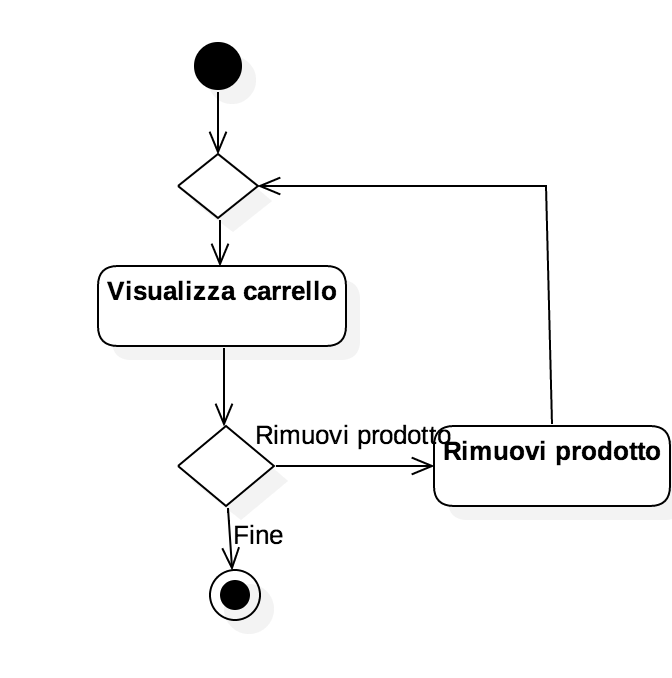
### Scheda di specifica

|  |  |
| --- | --- |
| ****Caso d’uso**** | **Visualizza carrello** |
| ID | UC2 |
| Attori | Cliente |
| Precondizioni | 1. Il cliente ha selezionato la visualizzazione del carrello |
| Postcondizioni | 1. Il cliente ha terminato l’aggiornamento del carrello |
| Sequenza degli eventi | 1. Il caso d’uso inizia quando il cliente seleziona il carrello 2. Il sistema mostra al cliente una finestra in cui è dettagliato il contenuto del carrello, indicando per ogni prodotto presente il relativo numero di pezzi. 3. Se il cliente clicca su rimuovi articolo:    1. L’articolo selezionato viene rimosso dal carrello 4. Se il cliente clicca su procedi all’acquisto:    1. Il cliente viene rimandato alla procedura di acquisto |

### Sequence diagram



### Activity diagram

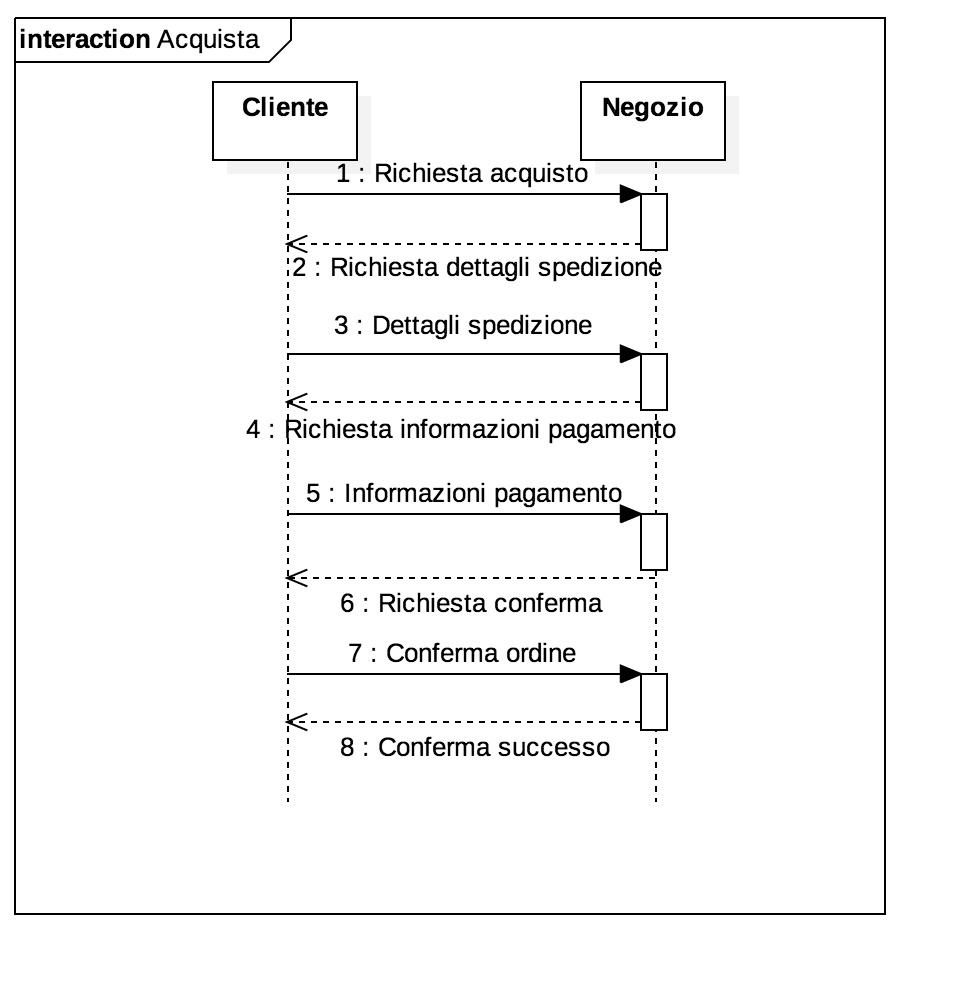


## Acquista

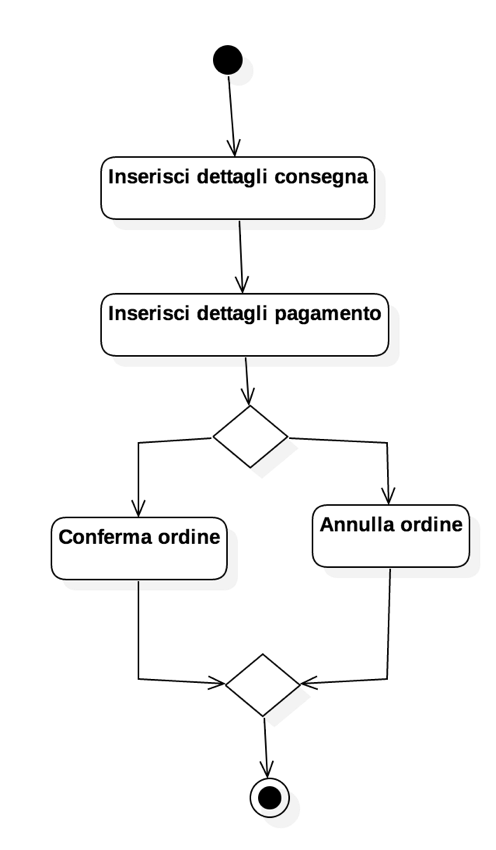
### Scheda specifica

|  |  |
| --- | --- |
| ****Caso d’uso**** | **Acquista** |
| ID | UC3 |
| Attori | Cliente |
| Precondizioni | 1. Il cliente ha eseguito l’accesso al negozio 2. Il cliente ha inserito nel carrello i prodotti che desidera acquistare 3. Il cliente ha cliccato su procedi all’acquisto |
| Postcondizioni | 1. Il cliente ha terminato l’acquisto |
| Sequenza degli eventi | 1. Il caso d’uso inizia quando il cliente desidera procedere all’acquisto 2. Il sistema mostra al cliente una finestra in cui inserire i dettagli relativi alla modalità di consegna del prodotto, quali indirizzo di consegna e modalità di spedizione da utilizzare 3. Il cliente inserisce tali dati e conferma cliccando su avanti 4. Il sistema calcola l’importo totale dovuto, tenendo conto di eventuali sconti di cui il cliente può usufruire 5. Il sistema richiede l’inserimento della modalità di pagamento dei prodotti 6. Il cliente inserisce tali dati 7. Il sistema mostra una finestra di riepilogo dell’ordine e chiede la conferma definitiva per procedere all’acquisto 8. Se l’utente conferma:    1. L’ordine viene registrato dal sistema e all’utente viene inviato un messaggio di conferma che l’ordine è avvenuto con successo |
| Sequenza alternativa | 1. Se il cliente decide di interrompere la procedura di acquisto:    1. Il sistema rimanda il cliente alla pagina iniziale    2. Il sistema mantiene i prodotti nel carrello di modo che possano essere acquistati in un secondo momento. |

### Sequence diagram



### Activity diagram

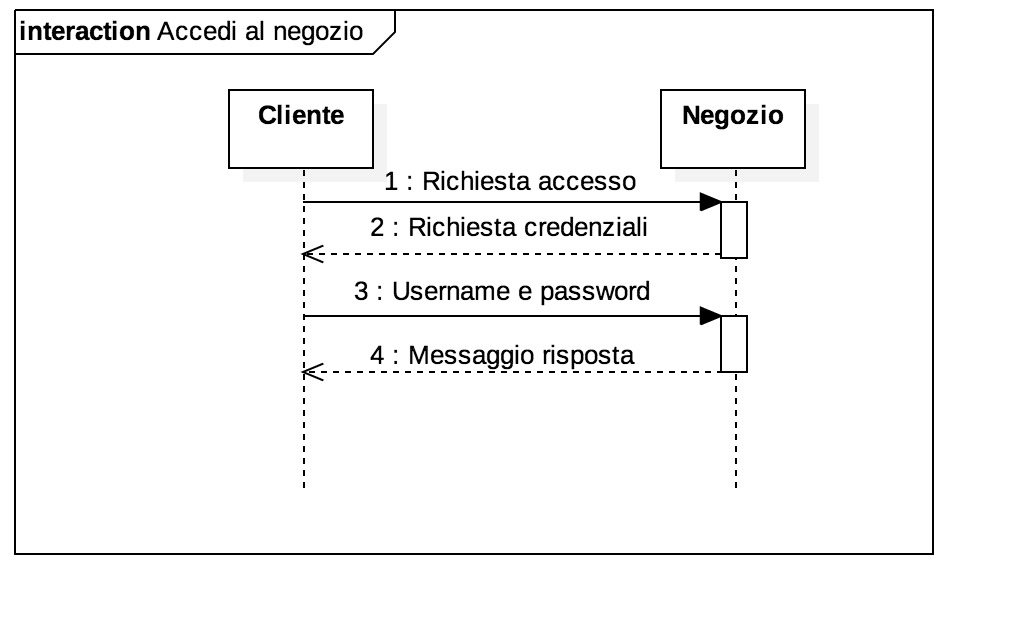


## Accedi al negozio

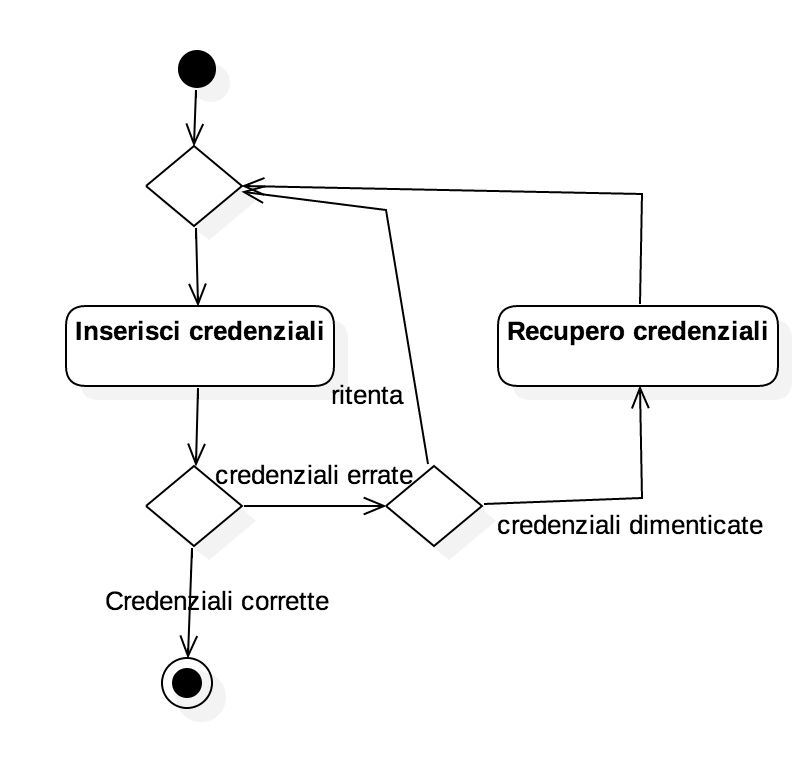
### Scheda di specifica

|  |  |
| --- | --- |
| ****Caso d’uso**** | **Accedi al negozio** |
| ID | UC4 |
| Attori | Cliente |
| Precondizioni | 1. Il cliente non è autenticato al sistema 2. Il cliente possiede già un account presso il sistema |
| Postcondizioni | 1. Il cliente è autenticato al sistema |
| Sequenza degli eventi | 1. Il caso d’uso inizia quando il cliente desidera effettuare l’accesso al negozio 2. Il sistema mostra al cliente una finestra in cui viene richiesto l’inserimento di username e password 3. Il cliente inserisce le sue credenziali e clicca su login 4. Se le credenziali sono corrette:    1. il cliente è autenticato al sistema |
| Sequenza alternativa | 1. Se il cliente inserisce delle credenziali errate:    1. il sistema lo invita a riprovare. 2. Se il cliente ha dimenticato le proprie credenziali:    1. il sistema mette a disposizione una procedura di ripristino delle credenziali mediante l’invio di un messaggio email. |

### Sequence diagram



### Activity diagram

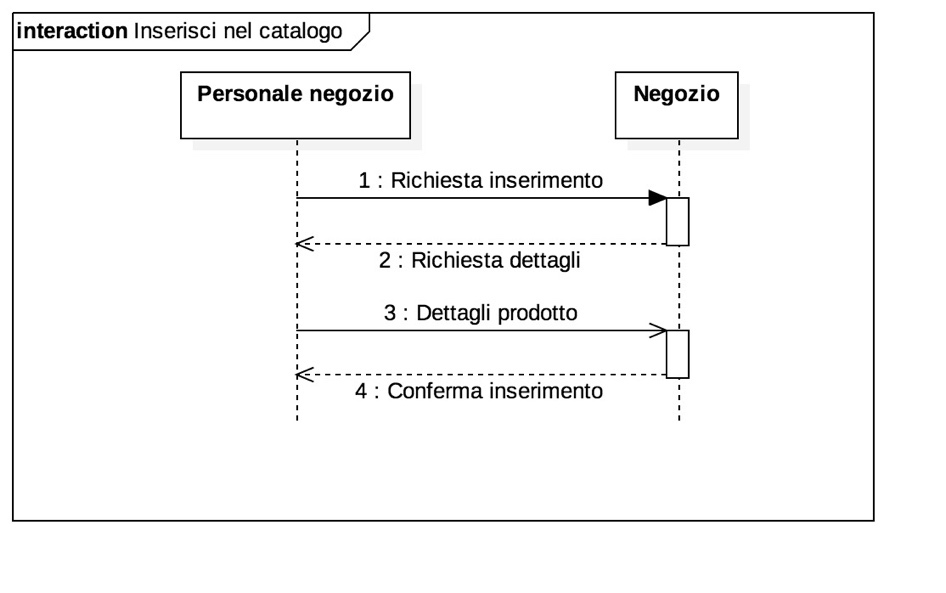


## Inserisci nel catalogo

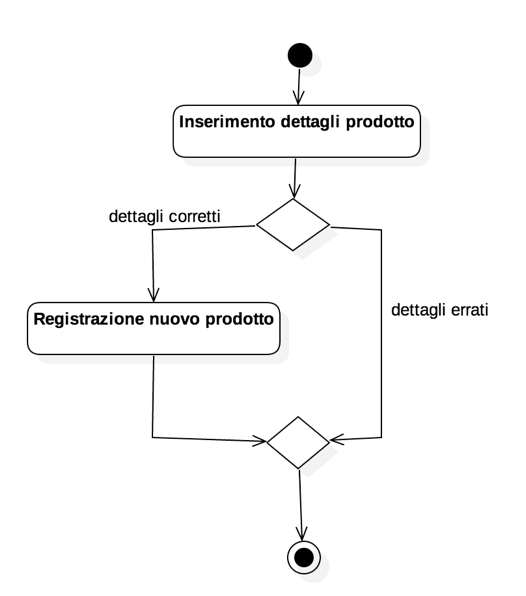
### Scheda specifica

|  |  |
| --- | --- |
| ****Caso d’uso**** | **Inserisci nel catalogo** |
| ID | UC5 |
| Attori | Personale negozio |
| Precondizioni | 1. Il personale del negozio desidera inserire nuovi CD/DVD a catalogo 2. Il personale del negozio ha effettuato l’accesso al sistema |
| Postcondizioni | 1. Il catalogo è stato aggiornato |
| Sequenza degli eventi | 1. Il caso d’uso inizia quando il personale del negozio intende inserire nuovi CD/DVD nel catalogo 2. Il sistema mostra una finestra in cui inserire i dettagli relativi al nuovo CD/DVD 3. Se i dati inseriti sono corretti:    1. il nuovo CD/DVD appare nel catalogo ed è acquistabile da parte del cliente |

### Sequence diagram



### Activity diagram



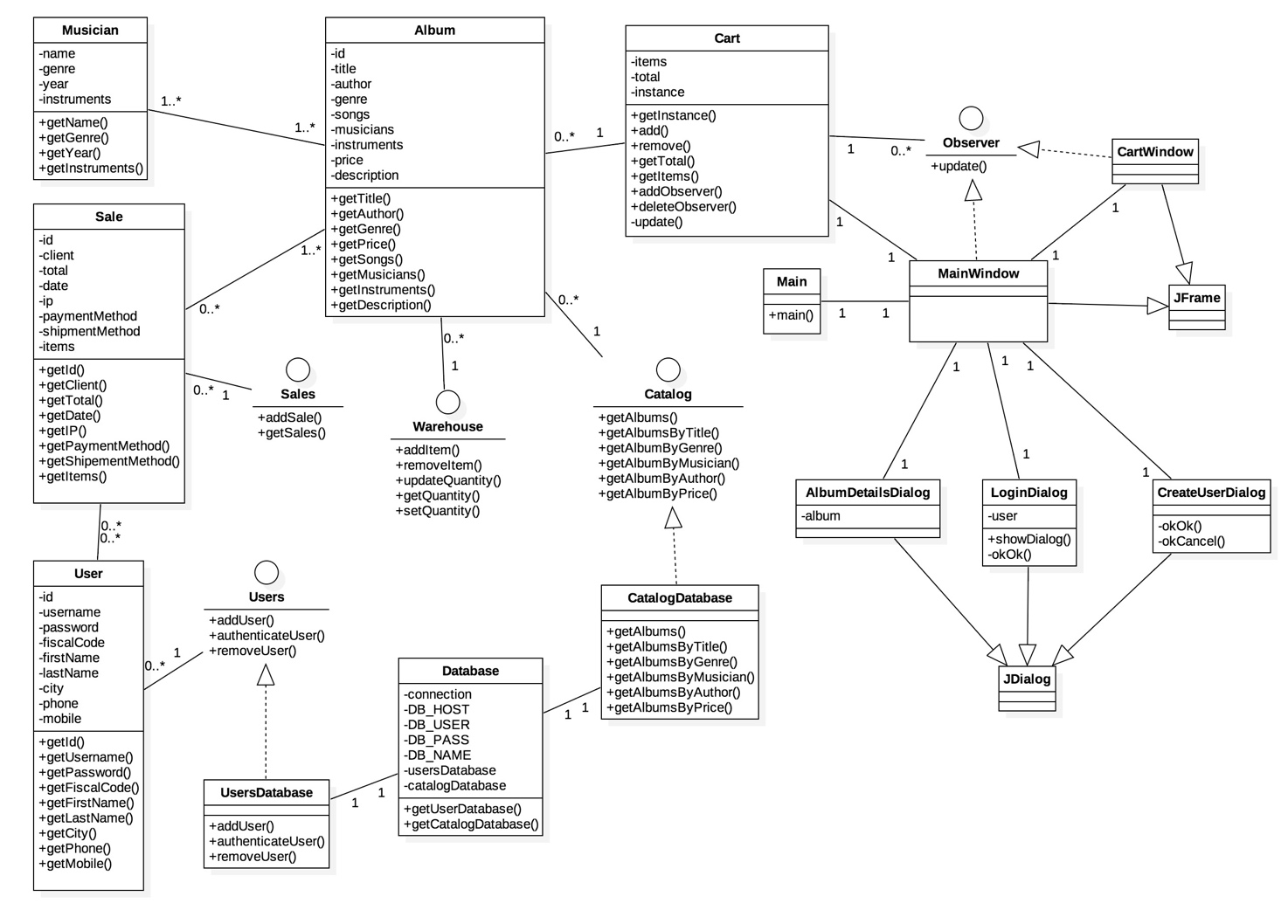
# Implementazione del prototipo

Per l’implementazione del prototipo ho scelto di utilizzare il linguaggio di programmazione Java 8 utilizzando per la realizzazione dell’interfaccia grafica il framework Swing.

Per salvare i contenuti dell’applicazione, ho utilizzato una base di dati relazionale, scegliendo PostgreSQL come DBMS.

Infine per gestire il codice sorgente ho utilizzato il sistema di VCS git, appoggiandomi su GitHub per l’hosting del repository.

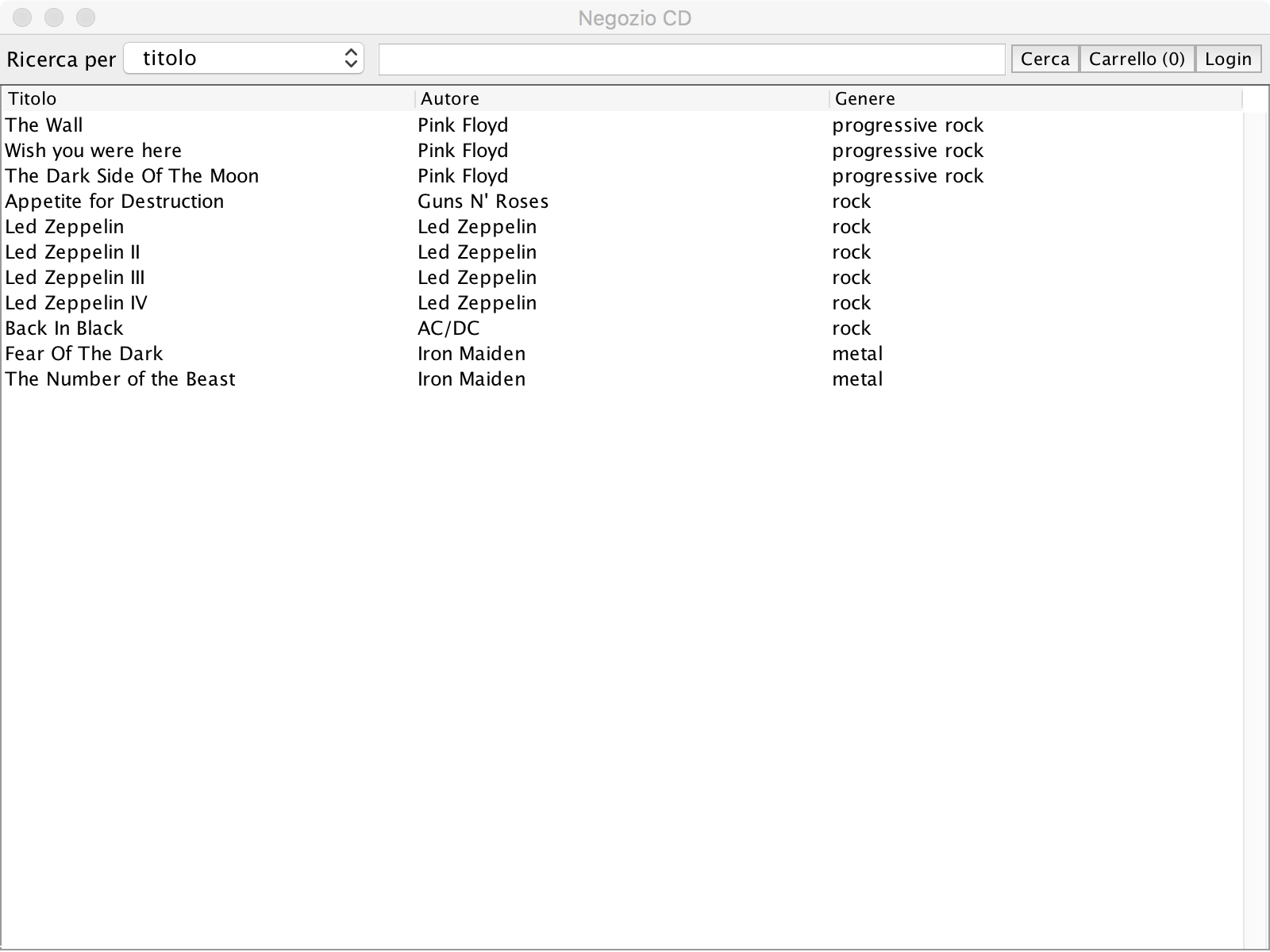
## Class diagram



(sono omessi il dettaglio dei campi delle classi relative all’interfaccia grafica per semplicità del diagramma)

## Interfaccia utente

### Finestra principale



### Visualizzazione dettagli album



### Visualizzazione carrello



### Finestre di login utenti

### 

### Finestra registrazione utenti

## 

## Design pattern utilizzati

Nell’implementazione ho utilizzato alcuni design pattern comuni:

* Pattern MVC (Model-View-Controller) – questo pattern consiste nel separare la parte di gestione dei dati (Model) dalla parte di presentazione dei dati all’utente (View) e dal controllo ossia la logica dell’applicazione (Controller). Tuttavia non ho creato esplicitamente una classe separata per il Controller ma ho mischiato tale codice con il codice della View, per semplificare l’applicazione.
* Pattern DAO (Data Access Object) – questo pattern serve per astrarre la base di dati di modo da renderne più facile l’accesso. Esso consiste nel creare delle classi che rappresentano i vari oggetti memorizzati nella base di dati, e delle interfacce per consentire l’accesso ai dati. Dopo di che le interfacce vengono implementate secondo lo specifico DBMS utilizzato, per cui se si volesse cambiare il modo di salvare i dati basterebbe solamente cambiare le interfacce.
* Pattern Observer – questo pattern consiste nel creare un interfaccia Observer con un metodo update() che viene richiamato quando l’oggetto osservato viene modificato. In questo caso ho utilizzato il pattern Observer per notificare a vari componenti dell’interfaccia grafica la modifica del contenuto del carrello.
* Pattern Singleton – questo pattern consiste nel creare un oggetto di cui può essere presente solo una singola istanza: per fare questo viene dichiarato privato il costruttore della classe e si crea un metodo getInstance() che ritorna l’istanza di tale oggetto.

# Testing del prototipo

Per testare il prototipo mi sono avvalso del framework JUnit 5 creando delle classi di test per i componenti dell’applicazione che riguardano l’accesso alla base di dati. Vengono testate le classi che consentono di accedere al database, di ottenere gli album dal database, e viene testata l’aggiunta, l’eliminazione e l’autenticazione di utenti al sistema.

Per il testing dell’interfaccia grafica invece ho invece testato l’applicazione su diversi sistemi operativi (macOS, Windows, Linux) per verificare che funzioni correttamente su tutti, e ho testato anche diverse impostazioni di DPI dello schermo per assicurarmi che gli elementi grafici venissero riprodotti correttamente su ogni tipo di display.