Delete Ordine

dbo. Dettaglio\_Ordine

dbo. Ordine

dbo.Utente

dbo. Prodotto

OrdineController

DeleteOrdine()

AcademyShopDB

RepositoryOrdine

IRepositoryOrdine

DeleteOrdineAsync()

DeleteOrdineAsync()

DeleteOrdineAsync()

ManageOrdineBusiness

ManageOrdineData

DeleteOrdineAsync()

# Requisiti funzionali metodo Delete

1. **Cancellazione ordine esistente:** il metodo deve accettare in input l’**ID** dell’utente che intende cancellare un ordine già esistente e l’**ID** del dettaglio dell’ordine che intente cancellare.

Successivamente sarà necessario effettuare i seguenti passaggi:

* + **Recupero ID ordine**: recupero dell’**ID** dell’ordine associato all’**ID** del dettaglio dell’ordine e all’**ID** dell’utente.
  + **Recupero stato ordine:** recupero dello stato dell’ordine associato all’**ID** dell’ordine recuperato precedentemente.
  + **Verifica stato ordine**: è necessario verificare che lo stato dell’ordine non sia **CHIUSO** (nel caso in cui l’ordine sia in stato **CHIUSO,** va restituito status code 400).
  + **Verifica della quantità di prodotto:** è necessario verificare che la quantità di prodotto recuperata dal Database sia <=a quella scelta dall’utente, eche sia diversa da 0 (nel caso in cui le citate condizioni non siano verificate, occorre restituire status code 400).
  + **Aggiornamento tabelle**: eseguire le seguenti operazioni sul Database:
    - * cancellazione del dettaglio ordine
      * cancellazione dell’ordine
      * aggiornamento della tabella PRODOTTO della quantità residua di prodotto, aumentata a seguito della cancellazione dell’ordine
  + **Ritorno dati e gestione errori**: nel caso in cui la cancellazione sul Database vada a buon fine, è necessario ritornare al consumer un JSON con i dati dell’ordine modificato, e status code 200. Nel caso in cui l’operazione di CRUD sul Database fallisca, occorre ritornare status code 500.

## ApiLayer : OrdineController.cs

A screen shot of a computer code

Description automatically generated

Metodo DeleteOrdine() presente nell’OrdineController.cs. Questo metodo va a richiamare il metodo DeleteOrdineAsync() situato nel BusinessLayer.

Come parametri prende in input idUtente ed idDettaglioOrdine per la cancellazione dell’ordine.

## BusinessLayer: ManageOrdineBusiness.cs

A screen shot of a computer program

Description automatically generated

Metodo DeleteOrdineAsync(), presente nel ManageOrderBusiness.cs. Questo metodo chiama tutte le funzioni presenti nel DataLayer effettuando tutti i controlli necessari nel caso in cui si riscontri un qualsiasi errore durante il processo di eliminazione dell’ordine.

### Verifica dell'Esistenza dell'Ordine:

* + **Codice**: **int? idOrdineEsistente = await oODL.RecuperaIdOrdineAsync(idUtente, idDettaglioOrdine);**
  + **Descrizione**: Controlla se l'ordine esiste utilizzando l'ID utente e l'ID dettaglio dell'ordine. Se l'ordine non esiste (**idOrdineEsistente == null**), restituisce un errore 400.

### Verifica dello Stato dell'Ordine:

* + **Codice**: **int statoOrdine = (int)await oODL.RecuperaStatoOrdineAsync((int)idOrdineEsistente);**
  + **Descrizione**: Recupera lo stato dell'ordine. Se lo stato dell'ordine è "chiuso" (**statoOrdine == 3**), restituisce un errore 400.

### Verifica dell'Esistenza del Prodotto:

* + **Codice**: **int? idProdotto = await oODL.RecuperaIdProdottoAsync((int)idOrdineEsistente);**
  + **Descrizione**: Controlla se il prodotto associato all'ordine esiste. Se il prodotto non esiste (**idProdotto == null**), restituisce un errore 404.

### Recupero della Quantità del Prodotto:

* + **Codice**: **int? quantitaProdottoDisponibile = await oODL.RecuperaQuantitaProdottoAsync((int)idProdotto);**
  + **Descrizione**: Recupera la quantità disponibile del prodotto associato all'ordine.

### Cancellazione dell'Ordine:

* + **Codice**: **bool successo = await oODL.DeleteOrdineAsync((int)idOrdineEsistente);**
  + **Descrizione**: Tenta di cancellare l'ordine. Se l'operazione ha successo (**successo** è true), restituisce un successo 204. In caso contrario, restituisce un errore 400.

### Gestione degli Errori:

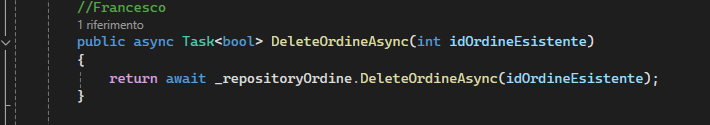
* + **Codice**: **catch (Exception)**
  + **Descrizione**: Gestisce eventuali eccezioni durante l'operazione di cancellazione, restituendo un errore 400 con un messaggio generico.

## DataLayer : ManageOrdineData.cs

A screen shot of a computer program

Description automatically generated

Metodi presenti nel ManageOrdineData.cs per il recupero di informazioni dal database.



Metodo DeleteOrdineAsync() che richiama il metodo DeleteOrdineAsync() dal RepositoryOrdine situato sempre nel DataLayer.

## Data Layer (IRepository) : IRepositoryOrdine.cs

A black background with white text

Description automatically generated

Interfaccia con il metodo DeleteOrdineAsync() che restituisce un booleano che accetta come parametro idOrdineEsistente.

## Data Layer (Repository) : RepositoryOrdine.cs

A computer screen shot of a program code

Description automatically generated

Metodo DeleteOrdineAsync() che prende come parametro idOrdineEsistente.

Questo metodo nel RepositoryOrdine.cs effettua le operazioni di query al Database per la cancellazione dell’ordine.