UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

CORSO DI LAUREA IN INFORMATICA

INSEGNAMENTO DI OBJECT ORIENTATION

A.A. 2024/2025

Progettazione e Sviluppo dell'applicativo Object Oriented

UninaFoodLab Chef Dashboard

A cura di

Antonio Esposito

Matricola N86005375

antonio.esposito167@studenti.unina.it

Docenti

Prof. Sergio Di Martino

Capitolo 1: Introduzione

Capitolo 2: Progettazione Software

Capitolo 1

Introduzione

1.1 Richiesta del committente

Si sviluppi un applicativo Java con interfaccia grafica (Swing o JavaFX) per la gestione dei corsi tematici offerti dalla piattaforma UninaFoodLab. Il sistema dovrà essere collegato a un database relazionale prepopolato contenente informazioni su chef, ricette e ingredienti.

Il sistema deve permettere l'autenticazione degli chef tramite credenziali (username e password). Una volta autenticato, lo chef può aggiungere un nuovo corso, specificando le seguenti informazioni: categoria, data di inizio, frequenza delle sessioni, numero di sessioni. Per ciascuna sessione, deve essere indicata la modalità di svolgimento, ovvero se si tratta di una sessione online o in presenza.

Lo chef avrà inoltre la possibilità di visualizzare i corsi esistenti, applicando filtri per categoria. Dopo aver selezionato un corso, lo chef può associare a ciascuna sessione pratica una o più ricette da realizzare.

Infine, il sistema deve fornire un report mensile, che permette allo chef di visualizzare: il numero di corsi totali tenuti, il numero di sessioni online e pratiche, e di quest'ultime il numero medio, massimo e minimo di ricette realizzate. Il report deve fornire una rappresentazione grafica dei dati, realizzata utilizzando una libreria come JFreeChart.

1.2 Breve descrizione dell'Applicazione

UninaFoodLab – Chef Dashboard è un'applicazione desktop con interfaccia in Java Swing, progettata per gli chef di un laboratorio culinario. Fornisce una dashboard centralizzata per gestire le proprie attività didattiche, tra cui la creazione di nuovi corsi di cucina, la gestione delle singole sessioni e l'associazione di ricette specifiche alle sessioni pratiche. L'applicazione offre anche strumenti di reportistica per monitorare le attività svolte in un dato periodo di tempo.

1.3 Panoramica delle Funzionalità

- **Autenticazione**: Accesso al sistema tramite credenziali (username/password).
- **Gestione Corsi**: Creazione di nuovi corsi, specificando categoria, data di inizio, numero e frequenza delle sessioni.
- **Gestione Sessioni**: Definizione della modalità (online o in presenza) per ogni sessione di un dato corso.
- **Associazione Ricette**: Collegamento di una o più ricette a sessioni pratiche.
- **Visualizzazione e Filtro**: Elenco di tutti i corsi tenuti dallo chef, con possibilità di filtrare per categoria.
- **Reportistica**: Generazione di un report mensile con statistiche aggregate e una visualizzazione grafica.

1.4 Tecnologie Utilizzate

- **Linguaggio**: Java 8+
- **GUI**: Java Swing
- **Grafici**: JFreeChart
- **Database**: PostgreSQL (gestito tramite JDBC)
- **Class Diagram**: Mermaid Charts Editor
- **Version Control:** https://github.com/AntoEsposito/UninaFoodLab.git

Capitolo 2

Progettazione Software

2.1 Pattern Architetturale: MVC

L'applicazione adotta un'architettura a più livelli, ispirata al pattern architetturale **MVC** (Model-View-Control), con una chiara separazione delle responsabilità.

Le classi sono state suddivise in packages in base alla loro responsabilità. Questa organizzazione modulare permette di avere un codice sorgente più facilmente leggibile e manutenibile nel tempo, semplificando inoltre la futura implementazione di nuove funzionalità.

Di seguito una breve descrizione dei principali packages nei queai è organizzato il codice dell'applicativo.

- **Model:** Rappresenta i dati e la logica di business. È suddiviso in:
  1. **Entity**: Oggetti che rappresentano le entità del dominio (es. `Corso`, `Chef`).
  2. **DAO (Data Access Objects)**: Classi responsabili dell'interazione con il database.
- **View**: L'interfaccia utente, costruita con classi Java Swing. È responsabile della presentazione dei dati e della cattura dell'input utente.
- **Control**: La classe Controller agisce da mediatore. Riceve gli input dalla View, elabora le richieste e aggiorna la View con i nuovi dati.

Questa architettura promuove un **basso accoppiamento** e una chiara **separazione delle responsabilità**. Le classi del modello (responsabili dei dati) operano in modo del tutto indipendente dalle viste (le interfacce utente), mentre il controller agisce come unico mediatore tra i componenti del sistema.

2.2 Model: Descrizione delle entità del problema

1. **Categoria**: Rappresenta la categoria tematica di un corso. È un'entità semplice che serve a classificare i corsi in base all'argomento (es: “Cucina Italiana). Ogni corso può appartenere a una sola categoria.
2. **FrequenzaSessioni**: Descrive la cadenza con cui si svolgono le lezioni di un corso (es: “Settimanale”, “Mensile”). Similmente a Categoria, è un'entità usata per classificare e definire una proprietà del Corso.
3. **Chef**: Rappresenta un cuoco/insegnante. Contiene le sue informazioni anagrafiche (nome, cognome) e di accesso (username, password). Ogni chef può gestire 0 o più corsi.
4. **Corso**: È una delle entità centrali del sistema e rappresenta un corso di cucina completo. Contiene tutte le informazioni principali come la data di inizio, il numero di lezioni, a quale Categoria appartiene, quale Chef lo tiene e, soprattutto, l'elenco di tutte le Sessioni (lezioni) che lo compongono.
5. **Sessione**: Rappresenta una singola lezione all'interno di un Corso. Contiene dettagli specifici della lezione come la data, se è in presenza o online (con relativo URL), a quale Corso appartiene e l'elenco delle Ricette che verranno trattate durante quella specifica lezione.
6. **Ricetta**: Rappresenta una singola ricetta che viene insegnata durante le lezioni. Ogni ricetta ha un nome e può essere preparata in una o più Sessioni di corsi diversi.

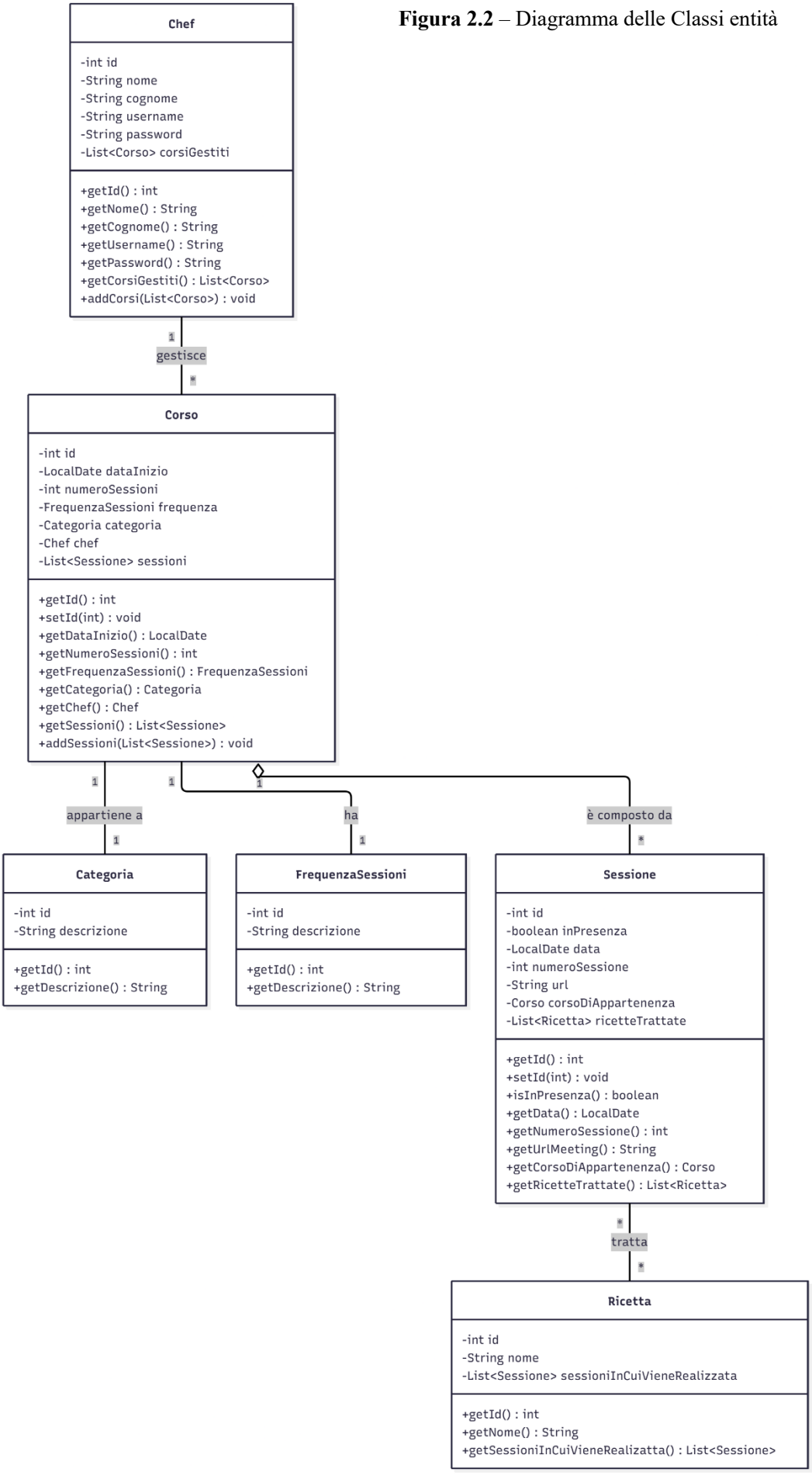

Figura 2.2 – Diagramma delle Classi entità

2.3 Model: Descrizione dei DAO

1. **DatabaseConnection**: Classe Singleton che centralizza la gestione delle connessioni al database **PostgreSQL**. Fornisce un punto di accesso unico e controllato per ottenere connessioni al database, garantendo coerenza nella configurazione (URL, credenziali) e facilitando eventuali modifiche future alla strategia di connessione.
2. **CategoriaDAO**: Gestisce l'accesso ai dati delle categorie di corsi culinari. Consente di recuperare le categorie disponibili nel sistema, sia singolarmente che in blocco, per permettere la classificazione e il filtraggio dei corsi.
3. **FrequenzaSessioniDAO**: Si occupa della gestione delle diverse cadenze temporali con cui possono svolgersi le sessioni dei corsi.
4. **RicettaDAO**: Gestisce il catalogo delle ricette culinarie presenti nel sistema. Permette di recuperare le ricette sia individualmente che in relazione alle sessioni in cui vengono trattate, fornendo i dati necessari per associare contenuti didattici specifici alle lezioni.
5. **ChefDAO**: Responsabile della gestione dei dati degli chef/insegnanti che tengono i corsi. Oltre a recuperare le informazioni anagrafiche degli chef, gestisce anche l'autenticazione tramite username e password. Carica in modo completo il profilo dello chef includendo tutti i corsi da lui tenuti.
6. **CorsoDAO**: Si occupa di creare nuovi corsi e recuperarli con tutti i dati associati (categoria, frequenza, chef, sessioni), permettendo filtraggio per chef e categoria. Coordina il caricamento di tutte le informazioni correlate per fornire una vista completa del corso.
7. **SessioneDAO**: Si occupa sia della creazione di nuove sessioni che del loro recupero completo di tutte le ricette associate.
8. **RealizzazioneRicettaDAO**: Gestisce la relazione molti-a-molti tra sessioni e ricette, implementando la tabella di associazione del database. Permette di collegare e scollegare ricette alle sessioni, gestendo quali ricette vengono trattate in quali lezioni, garantendo la flessibilità del sistema didattico. Questa è l'unica classe DAO che non ha un diretto corrispettivo nelle classi entity.

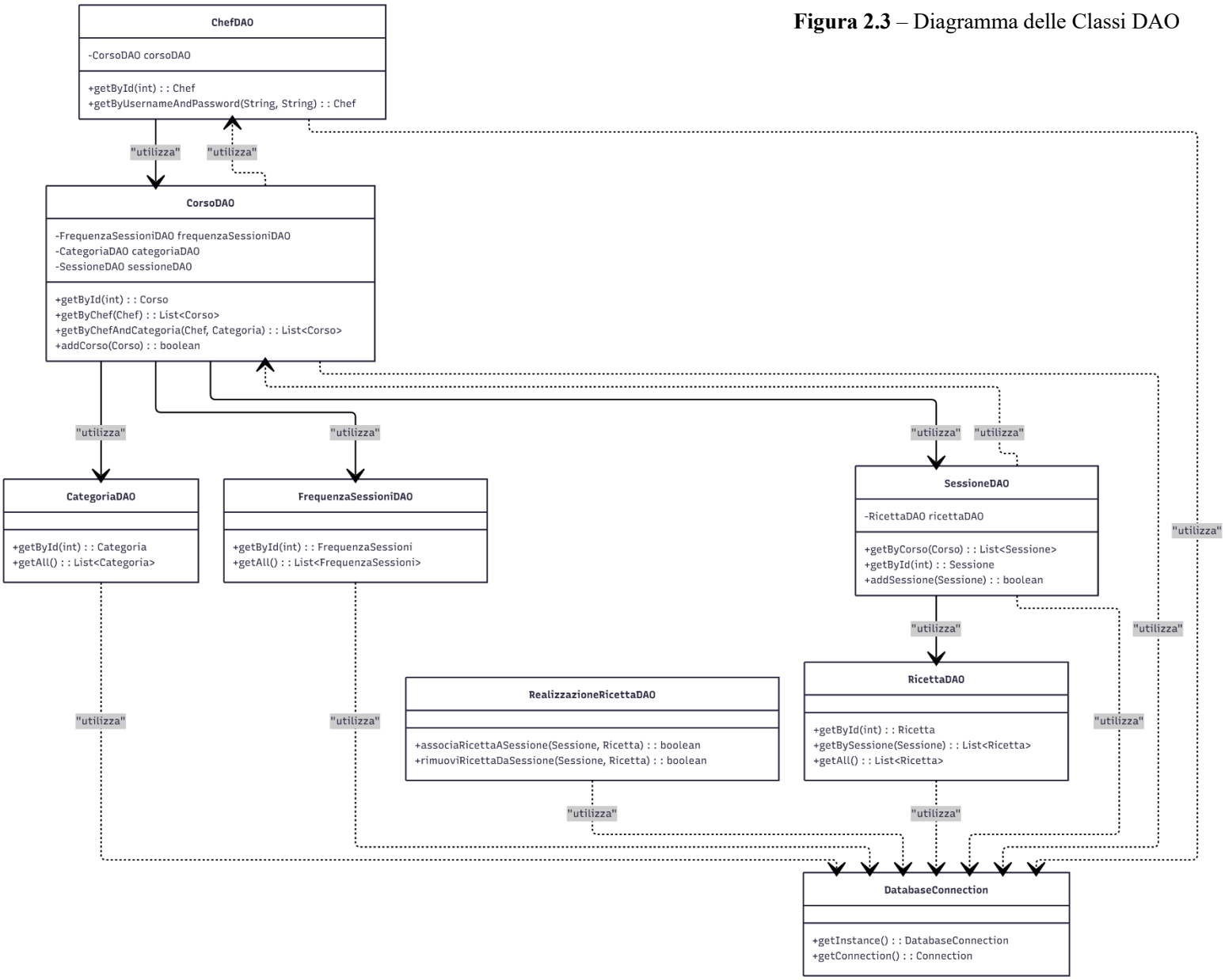

Figura 2.3 – Diagramma delle Classi DAO

Ogni classe DAO è responsabile della gestione della sua corrispondente classe entità ed è ovviamente in diretta relazione con quest'ultima.

Tuttavia, le classi entità sono state volutamente omesse dal diagramma per motivi di chiarezza, al fine di mostrare esclusivamente in che modo i DAO interagiscono tra loro.

2.4 Control: descrizione del Controller

Il Controller agisce come il cervello dell'applicazione: riceve input dalla vista, elabora le richieste utilizzando il modello e determina quale vista mostrare successivamente.

In particolare, il Controller si occupa di:

1. **Gestione della Logica di Business**: Contiene tutta la logica operativa dell'applicazione. Ad esempio, gestisce l'autenticazione dello chef, la creazione di nuovi corsi, l'associazione di ricette a una sessione, la generazione di report.
2. **Orchestrazione dei Dati**: Utilizza i DAO per recuperare, salvare e modificare i dati nel database.
3. **Controllo del flusso di esecuzione**: Gestisce la navigazione tra le diverse schermate. Ad esempio, dopo un login corretto, chiude la LoginPage e mostra la MainPage.
4. **Gestione dello Stato**: Mantiene in memoria lo stato corrente dell'applicazione, come lo chef che ha effettuato l'accesso (chefAutenticato) e il corso attualmente selezionato dall'utente (corsoSelezionato).
5. **Ponte tra View e Model**: Risponde alle interazioni dell'utente catturate dalla View (es. un click su un bottone), esegue le operazioni necessarie interagendo con il Model e, infine, aggiorna la View con i nuovi dati. Ad esempio, quando l'utente filtra i corsi per categoria, il Controller recupera i corsi filtrati tramite il CorsoDAO e li fornisce alla VisualizzaCorsiPage per aggiornare la relativa tabella.

Figura 2.4 – Classe Controller

2.5 View: Descrizione delle Interfacce

Ogni interfaccia è progettata per essere intuitiva e funzionale, comunicando esclusivamente con il Controller per eseguire operazioni e recuperare dati. Il tema e i colori sono semplici ed eleganti.

Di seguito una breve descrizione della GUI:

1. **LoginPage**: Schermata di accesso per l'autenticazione dello chef tramite username e password. È il punto di ingresso dell'applicazione.
2. **MainPage**: La dashboard principale a cui si accede dopo il login. Organizza le funzionalità in tre schede (Nuovo Corso, Visualizza Corsi, Report Mensile) e contiene il pulsante per il logout.
3. **NuovoCorsoPage**: Un form che permette allo chef di creare un nuovo corso, definendone i dettagli (categoria, data, numero e frequenza delle sessioni) e la modalità (online o in presenza) di ciascuna sessione.
4. **VisualizzaCorsiPage**: Mostra in una tabella l'elenco di tutti i corsi creati dallo chef. Permette di filtrare i risultati per categoria e di selezionare un corso per visualizzarne i dettagli.
5. **DettagliCorsoPage**: Finestra di dettaglio che mostra le informazioni e l'elenco delle sessioni di un singolo corso. Da qui è possibile accedere alla gestione delle ricette per le sessioni pratiche.
6. **AssociaRicettePage**: Interfaccia per gestire le ricette di una sessione pratica. Permette di associare ricette disponibili alla sessione o di rimuovere quelle già associate tramite due liste contrapposte.
7. **VisualizzaReportPage**: Pagina per la generazione di report mensili sull'attività dello chef. Mostra statistiche e un grafico a barre riassuntivo per il periodo selezionato.

Di seguito immagini dimostrative per dare un’idea del funzionamento di tutte le interfacce.

Figura 2.5.1 – Login Page

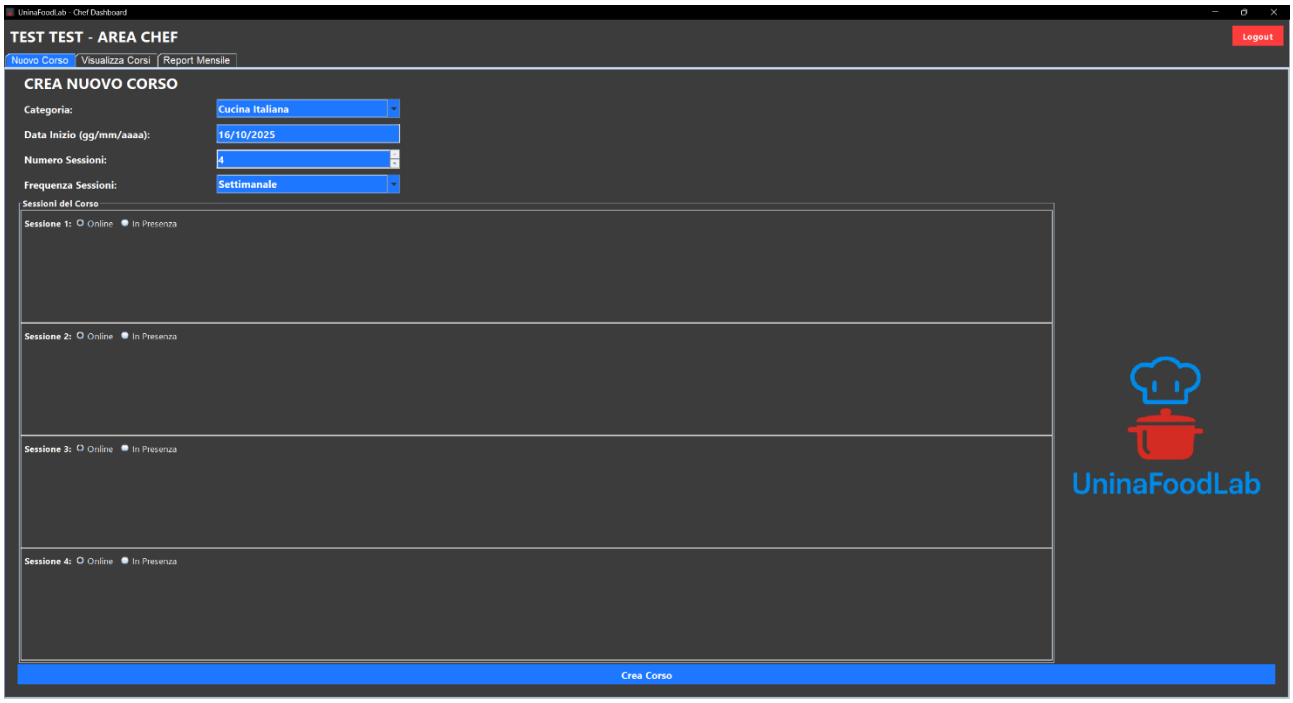

Figura 2.5.2 – Main Page, Tab Nuovo Corso

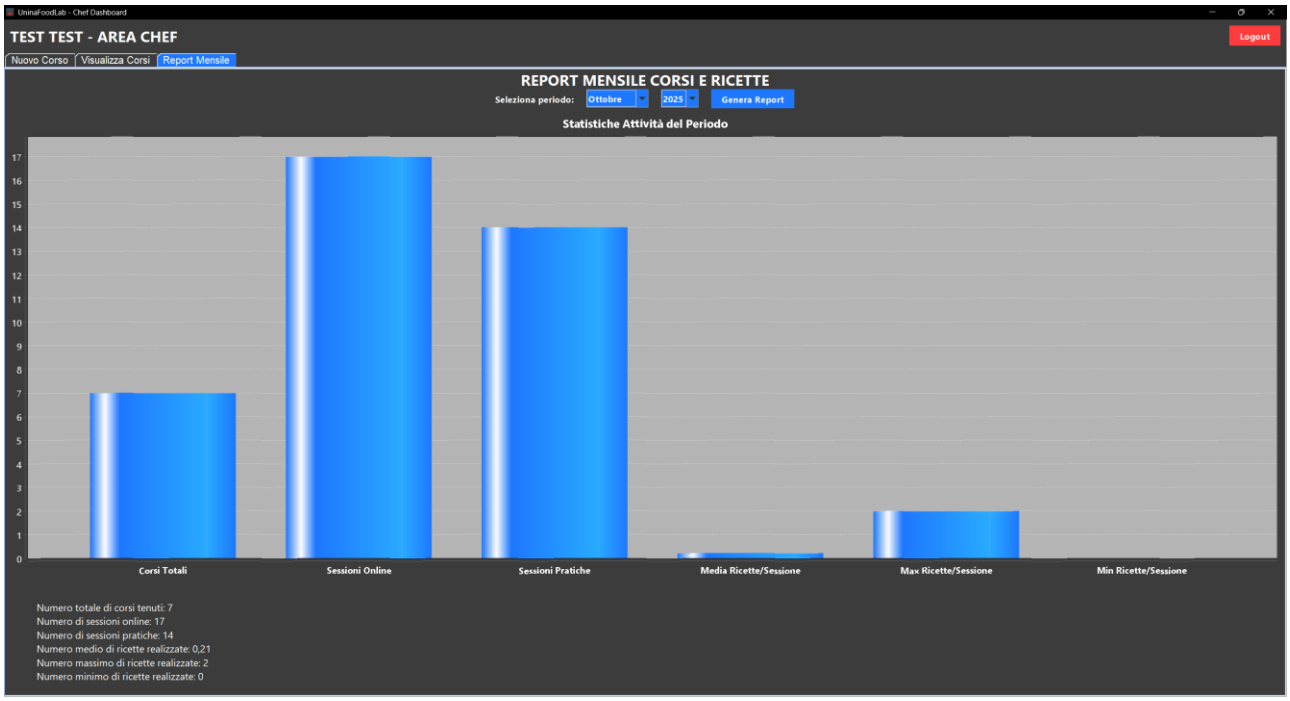

Figura 2.5.3 – Main Page, Tab Report Mensile

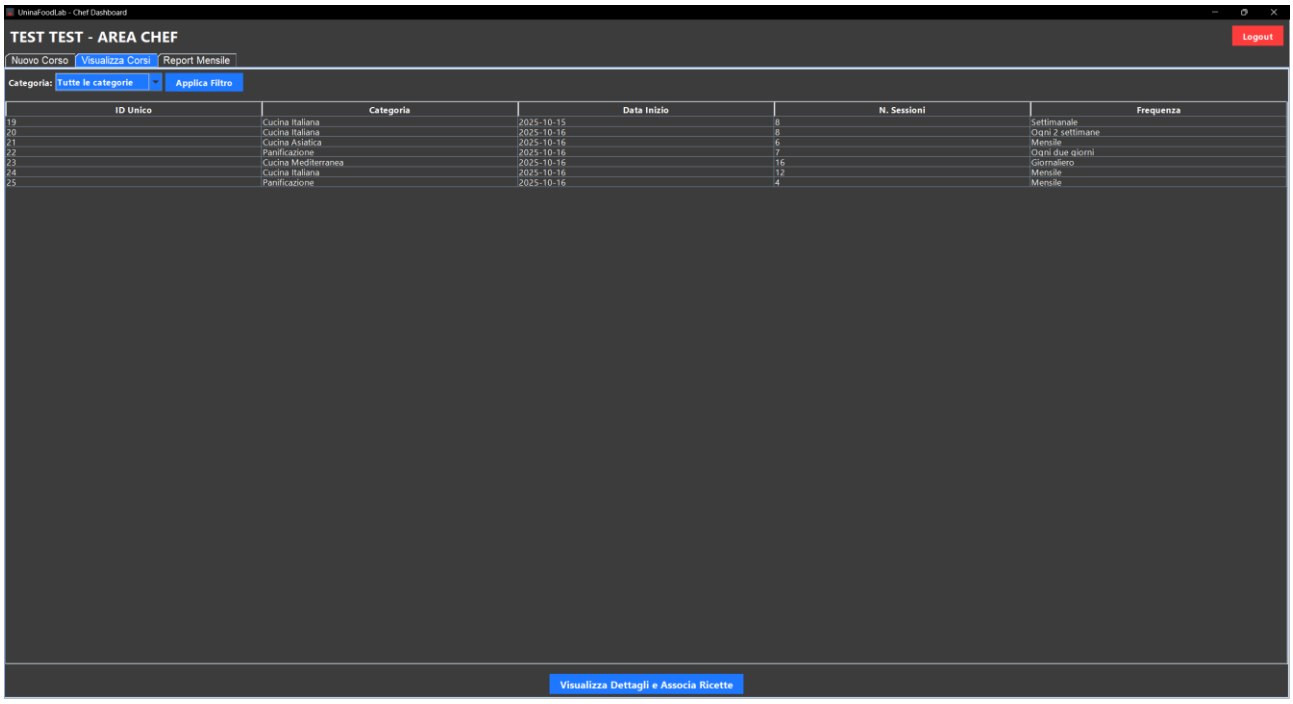

Figura 2.5.4 – Main Page, Tab Visualizza Corsi

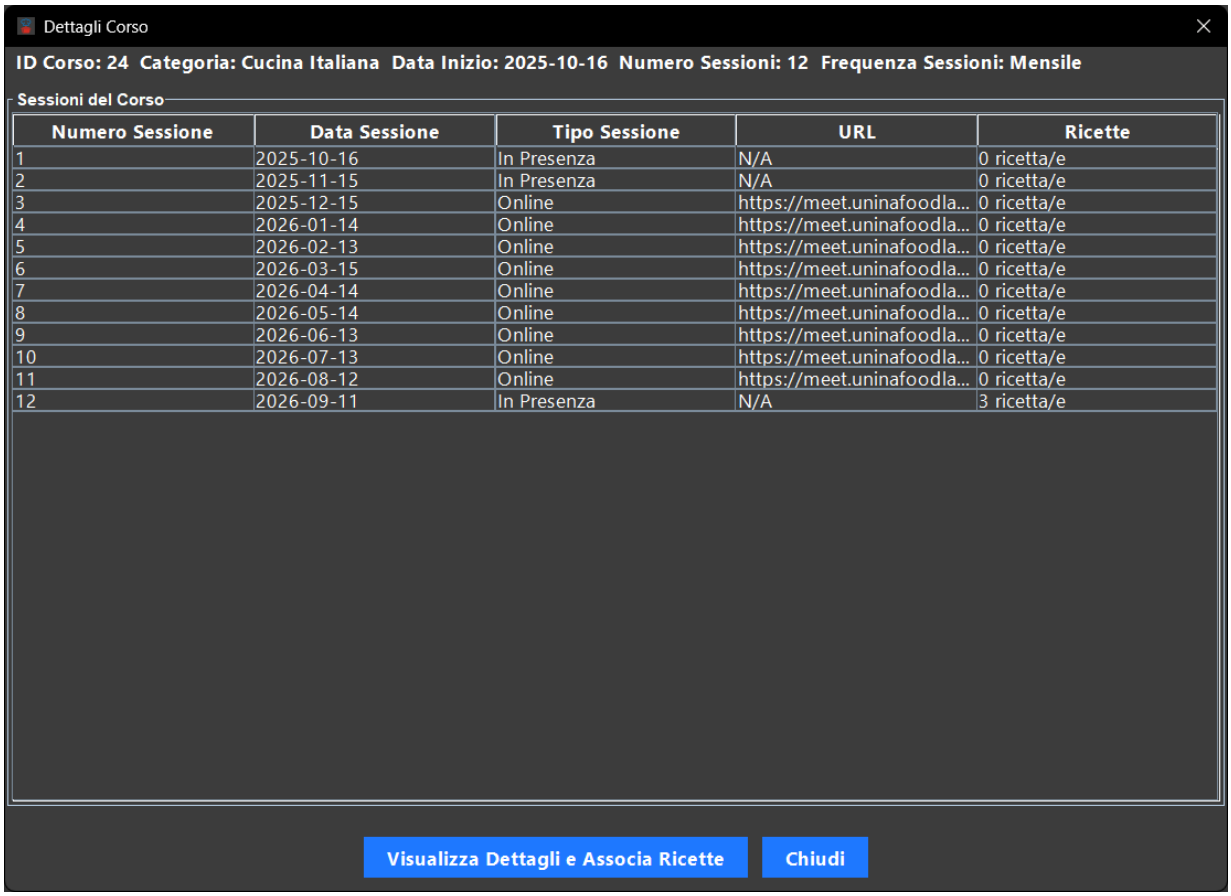

Figura 2.5.5 – Pagina Dettagli Corso

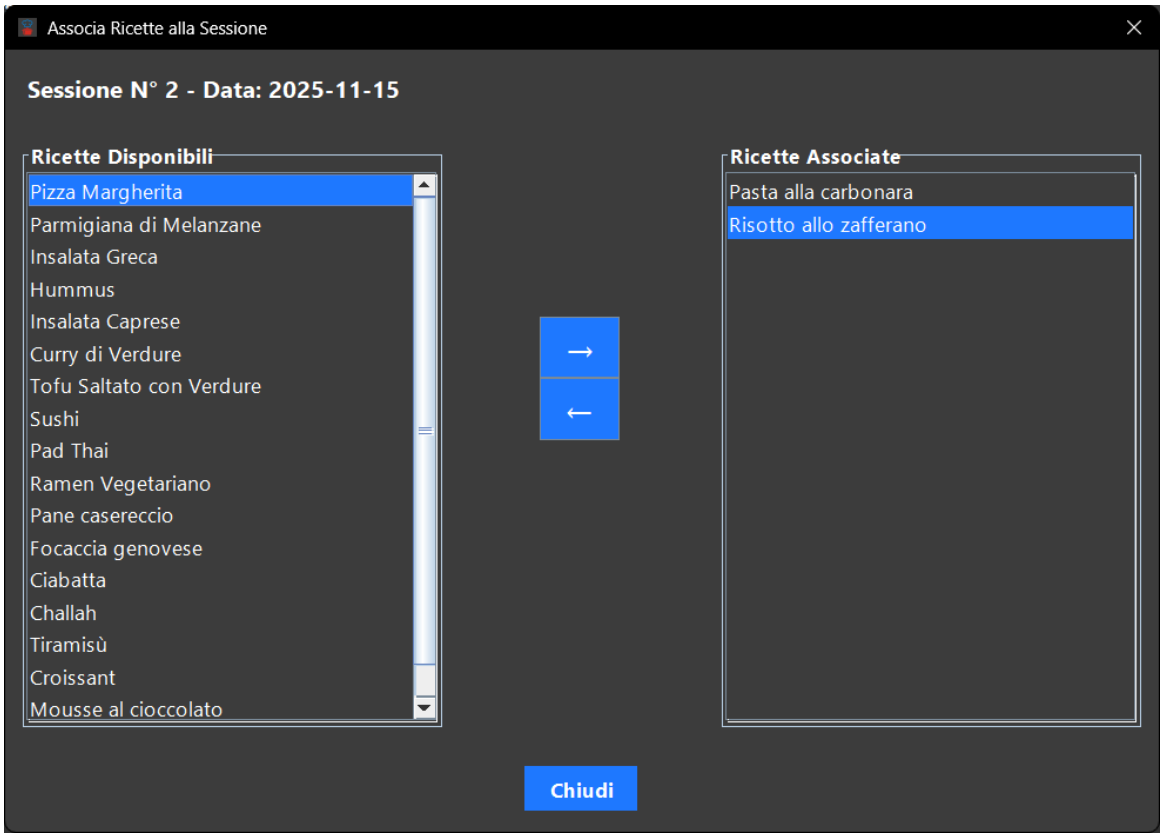

Figura 2.5.6 – Pagina Associazione Ricette

2.6 Class Diagram del sistema