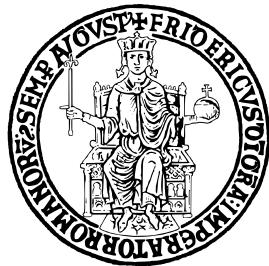


UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II



SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

CORSO DI LAUREA TRIENNALE IN INFORMATICA

OBJECT ORIENTATION

UNINAFOODLAB APPLICATION

Anno Accademico 2024–2025

Indice

1 Descrizione del progetto	3
1.1 Descrizione sintetica del problema	3
2 Diagramma delle Classi	4
2.1 UML	4
2.2 Architettura	4
3 Descrizione dei Package e delle Classi	6
3.1 Package <i>Boundary</i>	6
3.1.1 Classe LoginFrame	6
3.1.2 Classe HomepageChef	6
3.1.3 Classe NuovoCorsoFrame	6
3.1.4 Classe CorsiFrame	6
3.1.5 Classe SessioniFrame	6
3.1.6 Classe InserimentoSessioneFrame	6
3.1.7 Classe RicetteFrame	7
3.1.8 Classe NotificheFrame	7
3.1.9 Classe InvioNotificheFrame	7
3.1.10 Classe RiepilogoMensileFrame	7
3.2 Package <i>Control</i>	7
3.2.1 Classe Controller	7
3.3 Package <i>Entity</i>	7
3.3.1 Classe Chef	7
3.3.2 Classe Corso	7
3.3.3 Classe Notifica	8
3.3.4 Classe Prepara	8
3.3.5 Classe Ricetta	8
3.3.6 Classe Sessione	8
3.3.7 Classe SessioneInPresenza	8
3.3.8 Classe SessioneOnline	8
3.3.9 Classe Topic	8
3.4 Package <i>DaoInterface</i>	8
3.5 Package <i>DAO</i>	8
3.5.1 Classe ChefDAO	8
3.5.2 Classe CorsoDAO	9
3.5.3 Classe NotificaDAO	9
3.5.4 Classe PreparaDAO	9
3.5.5 Classe RicettaDAO	9
3.5.6 Classe SessioneOnlineDAO	9
3.5.7 Classe SessioneInPresenzaDAO	9

3.5.8	Classe TopicDAO	9
3.6	Package <i>DbConnection</i>	9
3.6.1	Classe DbConnection	10
4	Repository GitHub	11
4.1	Cos'è una Repository	11
4.2	Struttura del Progetto	11
4.3	Accesso al Codice	11

Capitolo 1

Descrizione del progetto

1.1 Descrizione sintetica del problema

Si sviluppi un applicativo Java con interfaccia grafica 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. Il sistema deve permettere allo chef di inserire delle notifiche relative ai propri corsi, in caso di modifiche, come cambio di data o ora di una sessione oppure la sua cancellazione. Durante la creazione della notifica, lo chef specifica se è destinata ad un singolo corso oppure se è risolta a tutti. Tutte le notifiche sono consultabili in un'apposita sezione dell'interfaccia.

Capitolo 2

Diagramma delle Classi

2.1 UML

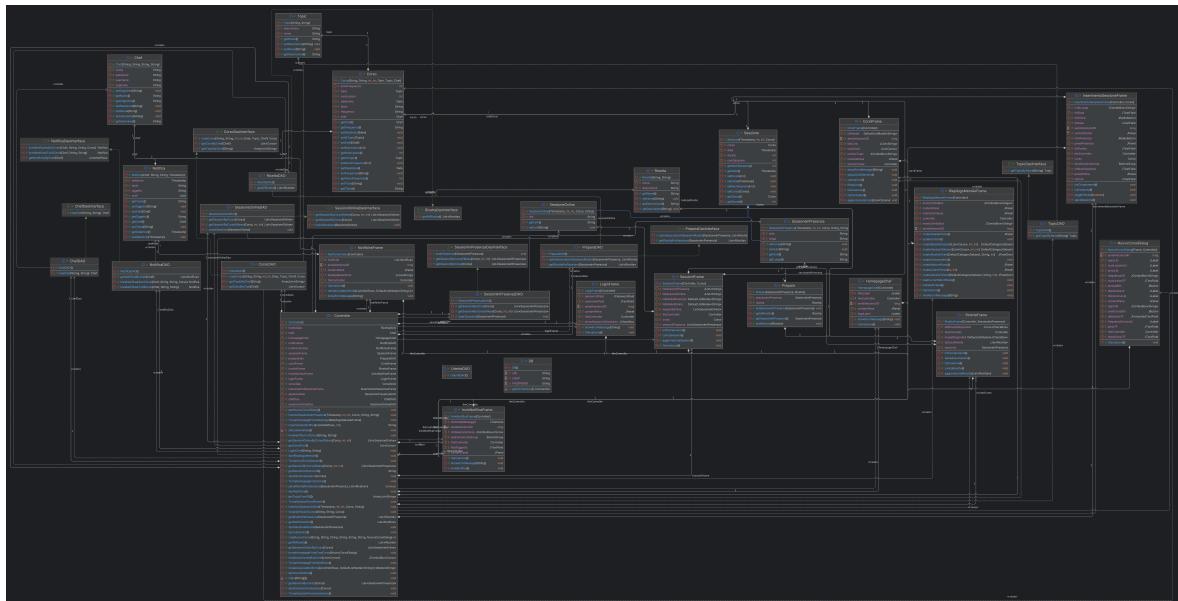


Figura 2.1: Diagramma delle classi UML

2.2 Architettura

L'Architettura dell'applicativo è basata sul pattern EBC (Entity Boundary Control). Questo pattern prevede la separazione delle responsabilità tra le diverse componenti del sistema, facilitando la manutenibilità e l'estensibilità dell'applicativo. Come si può vedere dal diagramma UML, le classi sono suddivise in tre categorie principali con ognuna di esse che svolge un ruolo specifico:

- **Entity:** Queste classi rappresentano le entità principali del dominio dell'applicativo, come Chef, Corso, Sessione, Ricetta, Ingrediente e Notifica. Esse contengono gli attributi e i metodi necessari per gestire i dati relativi a queste entità.

- **Boundary:** Le classi di questa categoria sono responsabili dell'interfaccia utente e della gestione delle interazioni con l'utente. Ad esempio, la classe UninaFoodLabApp funge da punto di ingresso per l'applicativo, mentre altre classi come LoginFrame, MainFrame e CourseManagementFrame gestiscono le diverse schermate dell'interfaccia grafica.
- **Control:** Queste classi agiscono come intermediari tra le classi Entity e Boundary. Gestiscono la logica di business dell'applicativo, coordinando le operazioni tra le entità e l'interfaccia utente. Ad esempio, la classe CourseController gestisce le operazioni relative ai corsi, mentre NotificationController si occupa della gestione delle notifiche.

Per l'interfaccia grafica è stato utilizzato il framework Swing di Java, che offre una vasta gamma di componenti per la creazione di interfacce utente ricche e interattive. Le classi del package Boundary estendono JFrame e utilizzano vari componenti Swing come JButton, JTextField, JComboBox e JTable per costruire le diverse schermate dell'applicativo. La connessione al database è gestita tramite JDBC (Java Database Connectivity), che consente di eseguire query SQL e di interagire con il database relazionale.

Capitolo 3

Descrizione dei Package e delle Classi

3.1 Package *Boundary*

Il pacchetto boundary contiene tutte le classi che regolano i Frame, la loro impostazione grafica, con disposizione dei widget e il loro funzionamento.

3.1.1 Classe LoginFrame

Questo è il frame per effettuare il login come chef nell'applicazione inserendo username e password validi.

3.1.2 Classe HomepageChef

Questo è il frame dell'HomePage dello chef dalla quale è possibile selezionare, grazie agli appositi buttoni, quale schermata visualizzare tra: Creazione corsi, Gestione Corsi, Notifiche e Riepilogo Mensile.

3.1.3 Classe NuovoCorsoFrame

Questo è il frame che permette la creazione di un nuovo corso

3.1.4 Classe CorsiFrame

Questo è il frame da cui è possibile visualizzare tutti i corsi dello chef loggato in formato di lista e filtrarli per Topic. Facendo doppio click sul nome di un corso è inoltre possibile aprire la schermata per l'inserimento di sessioni.

3.1.5 Classe SessioniFrame

Questo è il frame che permette di visualizzare e gestire le sessioni. Ogni sessione è selezionabile da una lista, selezionando una sessione in presenza verrà aperto il frame per la gestione delle ricette per quella sessione.

3.1.6 Classe InserimentoSessioneFrame

Questo è il frame che permette di creare le sessioni per il corso selezionato.

3.1.7 Classe RicetteFrame

Questo è il frame per la gestione delle ricette di una sessione in presenza, ad ogni sessione possono essere associate diverse ricette attraverso una checklist. Se una sessione ha già delle ricette associate un dialog chiede se si vuole che queste vengano sovrascritte.

3.1.8 Classe NotificheFrame

Questo è il frame dal quale è possibile visualizzare tutte le notifiche inviate dallo chef loggato e, attraverso l'apposito bottone, aprire il frame per l'invio di una nuova notifica.

3.1.9 Classe InvioNotificheFrame

Questo è il frame dal quale è possibile inviare nuove notifiche ad un corso specifico, selezionabile tramite ComboBox, oppure a tutti i corsi dello chef loggato.

3.1.10 Classe RiepilogoMensileFrame

Questo è il frame dal quale è possibile visualizzare il riepilogo mensile contenente: 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, sottoforma di diagramma a barre.

3.2 Package *Control*

Il pacchetto contiene la classe controller che gestisce la logica dell'applicativo determinando le interazioni tra le classi e la gestione delle risorse.

3.2.1 Classe Controller

Il controller, seguendo gli schemi dell'architettura BCE, si occupa di gestire le interazioni tra le interfacce del programma e le entità del dominio. Di conseguenza il controller ha il compito di amministrare l'apertura e la chiusura delle finestre della GUI, di interpretare l'interazione tra l'utente e gli elementi dell'interfaccia grafica, di gestire l'allocazione e la liberazione di risorse interagendo con il database mediante l'utilizzo delle classi DAO.

3.3 Package *Entity*

Il pacchetto contiene le classi che definiscono le entità del dominio.

3.3.1 Classe Chef

La classe Chef rappresenta un utente del sistema che può autenticarsi, gestire i propri corsi, creandone di nuovi o visualizzando quelli esistenti, e inserire notifiche relative ai corsi da lui tenuti. Lo Chef può, inoltre, gestire le sessioni dei propri corsi e le ricette delle sessioni in presenza.

3.3.2 Classe Corso

La classe Corso rappresenta i corsi tenuti dagli chef. Il Corso è caratterizzato da un titolo, una frequenza delle lezioni, una data di inizio, un anno in cui è tenuto il corso, un numero di lezioni, un topic e uno chef responsabile del corso.

3.3.3 Classe Notifica

La classe Notifica rappresenta le notifiche inviate dagli chef a corsi specifici e ricevute dagli studenti degli stessi corsi. La Notifica è caratterizzata da un oggetto, un testo e una data di invio.

3.3.4 Classe Prepara

La classe Prepara rappresenta la relazione tra le ricette e le sessioni in presenza. Di conseguenza gli attributi della classe sono quelli relativi alle ricette e alle sessioni associate.

3.3.5 Classe Ricetta

La classe Ricetta rappresenta le ricette che possono essere preparate durante le sessioni in presenza. La Ricetta è caratterizzata da un nome e una descrizione.

3.3.6 Classe Sessione

La classe Sessione rappresenta le lezioni, che compongono un corso. La Sessione è caratterizzata da una data, una durata, un numero e un corso di appartenenza.

3.3.7 Classe SessioneInPresenza

La classe SessioneInPresenza estende la classe Sessione e rappresenta le sessioni che si svolgono in presenza. La SessioneInPresenza è caratterizzata da una sede in cui si svolge la lezione e un'aula.

3.3.8 Classe SessioneOnline

La classe SessioneOnline estende la classe Sessione e rappresenta le sessioni che si svolgono online. La SessioneOnline è caratterizzata da un link di accesso.

3.3.9 Classe Topic

La classe Topic rappresenta le categorie dei corsi e le competenze degli chef. Il Topic è caratterizzato da un nome e una descrizione.

3.4 Package *DaoInterface*

Il package contiene le interfacce che definiscono i metodi per l'accesso ai dati nel database. Inoltre nel caso in cui si volesse cambiare il database, basterebbe creare una nuova classe DAO che implementa queste interfacce.

3.5 Package *DAO*

Il package contiene le classi che si occupano della connessione al database e dell'esecuzione delle query SQL.

3.5.1 Classe ChefDAO

La classe ChefDAO si occupa della gestione delle operazioni relative alla tabella Chef del database. Tramite il metodo creaChef di questa classe, è possibile istanziare un oggetto Chef a partire dai dati presenti nel database ed effettuare il login nell'applicativo.

3.5.2 Classe CorsoDAO

La classe CorsoDAO si occupa della gestione delle operazioni relative alla tabella Corso del database. Tramite i metodi di questa classe, è possibile aggiungere un nuovo corso, recuperare i corsi esistenti e applicare filtri per categoria.

3.5.3 Classe NotificaDAO

La classe NotificaDAO si occupa della gestione delle operazioni relative alla tabella Notifica del database.

Tramite i metodi di questa classe, è possibile aggiungere una nuova notifica, recuperare le notifiche esistenti e visualizzarle nell'apposita sezione dell'interfaccia.

3.5.4 Classe PreparaDAO

La classe PreparaDAO si occupa della gestione delle operazioni relative alla tabella Prepara del database.

Tramite i metodi di questa classe, è possibile associare una o più ricette a ciascuna sessione pratica di un corso.

3.5.5 Classe RicettaDAO

La classe RicettaDAO si occupa della gestione delle operazioni relative alla tabella Ricetta del database. Tramite i metodi di questa classe, è possibile recuperare le ricette esistenti e visualizzarle nell'apposita sezione dell'interfaccia.

3.5.6 Classe SessioneOnlineDAO

La classe SessioneOnlineDAO si occupa della gestione delle operazioni relative alla tabella SessioneOnline del database.

Tramite i metodi di questa classe, è possibile aggiungere una nuova sessione online o recuperare le sessioni online esistenti e visualizzarle nell'apposita sezione dell'interfaccia.

3.5.7 Classe SessioneInPresenzaDAO

La classe SessioneInPresenzaDAO si occupa della gestione delle operazioni relative alla tabella SessioneInPresenza del database.

Tramite i metodi di questa classe, è possibile aggiungere una nuova sessione in presenza o recuperare le sessioni in presenza esistenti e visualizzarle nell'apposita sezione dell'interfaccia.

3.5.8 Classe TopicDAO

La classe TopicDAO si occupa della gestione delle operazioni relative alla tabella Topic del database. Tramite i metodi di questa classe, è possibile recuperare i topic esistenti e visualizzarle nell'apposita sezione dell'interfaccia.

3.6 Package *DbConnection*

Il package contiene la classe che si occupa della connessione al database.

3.6.1 Classe DbConnection

La classe DbConnection si occupa di gestire la connessione al database. Fornisce un metodo statico `getConnection()` che restituisce un oggetto `Connection` per interagire con il database.

Capitolo 4

Repository GitHub

4.1 Cos'è una Repository

Una repository è uno spazio di archiviazione digitale dove vengono conservati i file di un progetto software insieme alla loro cronologia completa delle modifiche. GitHub è una piattaforma web che ospita repository Git, offrendo strumenti collaborativi per lo sviluppo software.

4.2 Struttura del Progetto

Il progetto UninaFoodLab è organizzato in una struttura chiara che facilita la navigazione e la comprensione del codice:

- **Codice sorgente:** Organizzato in package secondo il pattern architettonale EBC
- **Documentazione:** Include diagrammi UML, specifiche tecniche e guide utente
- **Database:** Script SQL per la creazione e il popolamento del database
- **Risorse:** Immagini, icone e altri file multimediali utilizzati dall'applicazione

4.3 Accesso al Codice

Il codice sorgente del progetto è disponibile al seguente indirizzo:

<https://github.com/xFireFox27/UninaFoodLab.git>

La repository contiene l'intero progetto con la cronologia completa delle modifiche, permettendo di seguire l'evoluzione del software durante lo sviluppo.