UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

CORSO DI LAUREA IN INFORMATICA

INSEGNAMENTO DI OBJECT ORIENTATION

ANNO ACCADEMICO 2024/2025

Documentazione per

Diagramma delle classi di design

Creazione e gestione codice sorgente Java (e Java Swing)

Autore:

Salvatore DE PASQUALE
MATRICOLA N86005746
salvat.depasquale@studenti.unina.it

Docente:

Prof. Sergio DI MARTINO

Contents

1 Introduzione

Questo documento descrive l'architettura e l'implementazione dell'applicazione Java per il progetto UninaFoodLab. L'applicazione, realizzata con Java e la libreria Swing per l'interfaccia grafica, permette agli utenti (Chef e Utenti) di interagire con un sistema di gestione per corsi di cucina. Le funzionalità principali includono la creazione e gestione di corsi, ricette e sessioni da parte degli Chef, l'iscrizione ai corsi e la conferma di partecipazione alle sessioni pratiche da parte degli Utenti. Lo Chef dispone inoltre di una sezione dove può calcolare, in base al numero di adesioni alla sessione, la quantità precisa di quanti ingredienti serviranno per tutti gli utenti iscritti. L'interazione con il database PostgreSQL è gestita tramite JDBC.

2 Struttura del Progetto Java

Il codice sorgente è organizzato in package secondo un sistema multi-strato (con DAO e Service) per promuovere la modularità, la separazione delle responsabilità e la manutenibilità.

- it.uninafoodlab.model: Contiene le classi Java che rappresentano le entità del dominio applicativo. Queste classi fungono da contenitori di dati.
- it.uninafoodlab.dao: Contiene le classi DAO (Data Access Object), responsabili dell'incapsulamento della logica di accesso ai dati persistenti nel database PostgreSQL. Utilizzano JDBC per eseguire query SQL.
- it.uninafoodlab.util: Contiene la classe per la gestione delle connessioni JDBC.
- it.uninafoodlab.service: Contiene le classi Service, che implementano la logica. Coordinano le operazioni, utilizzano i DAO per interagire con i dati e forniscono un'interfaccia di alto livello per il layer di presentazione (view).
- it.uninafoodlab.view: Contiene le classi che costituiscono l'interfaccia grafica utente, implementata con Java Swing. Include le finestre principali (JFrame) e le finestre di dialogo modali (JDialog).

3 Package it.uninafoodlab.model

Le classi in questo package sono semplici contenitori di dati con attributi privati e metodi getter/setter pubblici. Rappresentano le informazioni gestite dall'applicazione.

3.1 Adesione

Rappresenta lo stato di partecipazione di un Utente a una SessioneInPresenza.

- usernameUtente: String - Identificativo dell'utente.
- codFiscChef: String - Identificativo dello chef che tiene il corso.
- titoloCorso: String - Titolo del corso.

- titoloSessione: String - Titolo della sessione specifica.
- dataAdesione: java.sql.Timestamp - Data e ora di creazione/modifica dell'adesione.
- confermata: Boolean - Flag che indica se la partecipazione è confermata (true) o meno (false).

3.2 Chef

Rappresenta un utente con privilegi di Chef.

- codiceFiscale: String - Identificativo univoco dello Chef.
- nome: String
- cognome: String
- email: String - Indirizzo email (usato anche per login).
- passHash: String - Password per l'autenticazione.

3.3 ComposizioneRicetta

Definisce la presenza e la quantità di un Ingrediente all'interno di una Ricetta.

- codFiscChef: String
- titoloRicetta: String
- dataCreazioneRicetta: java.time.LocalDateTime
- nomeIngrediente: String - Nome dell'ingrediente utilizzato.
- quantitaIngrediente: double - Quantità numerica.
- unitaSpecIngrediente: String - Unità di misura (es. "g", "ml").

3.4 Corso

Rappresenta un corso offerto da uno Chef.

- codFiscChef: String - Identificativo dello chef proprietario.
- titolo: String - Titolo univoco del corso per quello chef.
- frequenza: String - Descrizione della frequenza (es. "settimanale").
- dataInizio: java.time.LocalDate - Data di inizio del corso.

3.5 Ingrediente

Rappresenta un ingrediente alimentare di base.

- nomeIngrediente: String - Nome univoco dell'ingrediente.
- descrizioneIngrediente: String - Descrizione opzionale.

3.6 Iscrizione

Rappresenta l'associazione tra un Utente e un Corso a cui è iscritto.

- usernameUtente: String
- codFiscChef: String
- titoloCorso: String
- dataIscrizione: java.sql.Timestamp - Data e ora dell'iscrizione.

3.7 ProgrammaSessione

Associa una Ricetta a una SessioneInPresenza, specificando le porzioni per partecipante.

- codFiscChef: String
- titoloRicetta: String
- dataCreazioneRicetta: java.time.LocalDateTime
- titoloCorso: String
- titoloSessione: String
- porzioniPerPartecipante: double - Numero di porzioni della ricetta da preparare per ogni partecipante confermato.

3.8 Ricetta

Descrive una ricetta creata da uno Chef.

- codFiscChef: String
- titoloRicetta: String
- descrizione: String - Testo descrittivo della ricetta.
- difficolta: String - Livello di difficoltà (es. "facile").
- tempoPrep: int - Tempo di preparazione stimato in minuti.
- porzioni: int - Numero di porzioni standard prodotte dalla ricetta.
- dataCreazione: java.time.LocalDateTime - Data e ora di creazione della ricetta.

3.9 SessioneInPresenza

Rappresenta una lezione o evento di un Corso che si tiene in un luogo fisico.

- codFiscChef: String
- titoloCorso: String
- titoloSessione: String - Titolo univoco della sessione per quel corso.
- dataOra: java.time.LocalDateTime - Data e ora di svolgimento.
- durata: int - Durata in minuti.
- luogo: String - Indirizzo o nome del luogo.
- postiTotali: int - Capienza massima.
- postiDisponibili: int - Posti attualmente liberi.

3.10 SessioneOnline

Rappresenta una lezione o evento di un Corso che si tiene online.

- codFiscChef: String
- titoloCorso: String
- titoloSessione: String - Titolo univoco della sessione per quel corso.
- dataOra: java.time.LocalDateTime - Data e ora di svolgimento.
- durata: int - Durata in minuti.
- linkVideo: String - URL per accedere alla sessione online.

3.11 Utente

Rappresenta un utente standard della piattaforma (non Chef).

- username: String - Identificativo univoco dell'utente.
- nome: String
- cognome: String
- email: String - Indirizzo email.
- passHash: String - Password per l'autenticazione.

4 Package it.uninafoodlab.dao

Questo package implementa i file DAO. Ogni classe è dedicata alla gestione della persistenza per una specifica classe del package model.

Responsabilità Principali:

- Eseguire operazioni CRUD (Create, Read, Update, Delete) sulla tabella corrispondente nel database PostgreSQL.
- Utilizzare l'API JDBC (java.sql.*) per la connessione e l'esecuzione delle query.
- Impiegare PreparedStatement per prevenire SQL Injection e migliorare le prestazioni.
- Mappare i dati tra gli oggetti ResultSet (risultati delle query) e gli oggetti del package model.
- Gestire le eccezioni SQLException.
- Ottenere connessioni tramite la classe DBConnection e chiuderle correttamente usando try-with-resources.

Classi DAO Implementate:

- AdesioneDAO: Metodi per inserire, leggere, aggiornare (updateConferma) e contare (countPartecipantiConfermati) le adesioni.
- ChefDAO: Metodi per inserire, leggere (per PK o email), aggiornare ed eliminare Chef.
- ComposizioneRicettaDAO: Metodi per aggiungere, leggere (singola o tutte per ricetta), aggiornare (quantità/unità) e rimuovere ingredienti da una ricetta.
- CorsoDAO: Metodi per inserire, leggere (singolo, tutti, o per chef), aggiornare ed eliminare Corsi.
- IngredienteDAO: Metodi per inserire, leggere (per nome o tutti), aggiornare ed eliminare Ingredienti base.
- IscrizioneDAO: Metodi per inserire, leggere (singola o tutte per utente) ed eliminare Iscrizioni.
- ProgrammaSessioneDAO: Metodi per aggiungere, leggere (singolo o tutto per sessione) e rimuovere ricette dal programma di una sessione.
- RicettaDAO: Metodi per inserire, leggere (per PK o tutte per chef), aggiornare ed eliminare Ricette.
- SessioneInPresenzaDAO: Metodi per inserire, leggere (singola, tutte, o per corso), aggiornare ed eliminare Sessioni In Presenza.
- SessioneOnlineDAO: Metodi per inserire, leggere (singola, tutte, o per corso), aggiornare ed eliminare Sessioni Online.
- UtenteDAO: Metodi per inserire, leggere (per username/email o tutti), aggiornare, eliminare Utenti e validare il login.

5 Package it.uninafoodlab.util

Contiene classi di utilità generale.

5.1 DBConnection

Fornisce un punto centralizzato per ottenere connessioni al database PostgreSQL.

- Contiene i parametri di connessione (URL, USER, PASSWORD) come costanti private statiche.
- Esegue il caricamento del driver JDBC org.postgresql.Driver in un blocco statico.
- Offre il metodo pubblico statico getConnection() che restituisce una nuova istanza di java.sql.Connection ad ogni chiamata, utilizzando DriverManager.getConnection().
- Non implementa pattern Singleton per la connessione, affidando la gestione del ciclo di vita ai chiamanti (i DAO).

6 Package it.uninafoodlab.service

Questo package contiene il cuore della logica applicativa, disaccoppiando la View dai dettagli dell'accesso ai dati.

6.1 ChefService

Incapsula la logica di business relativa alle azioni dello Chef.

- Mantiene un riferimento all'oggetto Chef autenticato.
- Contiene istanze di tutti i DAO necessari per le operazioni dello Chef.
- Fornisce metodi di alto livello che orchestrano chiamate ai DAO, ad esempio:
  - getMieiCorsi(), creaNuovoCorso(), aggiornaCorso(), eliminaCorso().
  - getMieRicette(), creaNuovaRicetta(), aggiornaRicetta(), eliminaRicetta().
  - getSessioni...DelCorso(), creaNuovaSessione...(), aggiornaSessione...(), eliminaSessione...().
  - getIngredientiDellaRicetta(), aggiungiIngredienteARicetta(), aggiornaIngredienteInRicetta(), rimuoviIngredienteDaRicetta(), creaNuovoIngrediente().
  - getProgrammaDellaSessione(), aggiungiRicettaAlProgramma(), rimuoviRicettaDalProgramma().
  - calcolaIngredientiPerSessione(): Logica chiave per il report, che interroga AdesioneDAO, ProgrammaSessioneDAO e ComposizioneRicettaDAO per calcolare il fabbisogno aggregato.

6.2 UserService

Incapsula la logica di business relativa alle azioni dell'Utente standard.

- Mantiene un riferimento all'oggetto Utente autenticato.
- Contiene istanze dei DAO necessari per le operazioni dell'Utente.
- Fornisce metodi di alto livello, ad esempio:
  - getTuttiICorsi(), getCorsiDisponibili(), getLeMieIscrizioni(), getMieiCorsiIscritto().
  - iscrivitiACorso(): Oltre a registrare l'iscrizione, crea anche le Adesione preliminari (con confermata=false) per le sessioni in presenza.
  - disiscrivitiDaCorso().
  - getSessioni...DelCorso().
  - getMiaAdesionePerSessione(), aggiornaConfermaAdesione(): Permettono di leggere e modificare lo stato di conferma per una sessione pratica.

7 Package it.uninafoodlab.view

Implementa l'interfaccia utente grafica utilizzando Java Swing. Le classi in questo package interagiscono con l'utente e utilizzano i Service per eseguire le operazioni richieste.

7.1 Classi Principali

- Main: Contiene il metodo main() che avvia l'applicazione creando e rendendo visibile la LoginFrame all'interno dell'Event Dispatch Thread (EDT) di Swing tramite SwingUtilities.invokeLater.
- LoginFrame (JFrame): Schermata iniziale per l'autenticazione. Utilizza JTextField, JPasswordField, JCheckBox e JButton. Chiama UtenteDAO o ChefDAO per la validazione. In caso di successo, istanzia il UserService o ChefService appropriato e apre la UserMainFrame o ChefMainFrame, passando l'oggetto utente/chef e il service.
- RegisterUserFrame (JFrame): Permette la registrazione di nuovi utenti standard. Utilizza JTextField, JPasswordField, JButton.
- ChefMainFrame (JFrame): Dashboard dello Chef. Strutturata con JTabbedPane ("Gestione Corsi", "Gestione Ricette", "Report Ingredienti"). Ogni tab contiene pannelli con JList, JButton, JComboBox, JTextArea ecc. Utilizza DefaultListModel per gestire dinamicamente il contenuto delle liste. Aggiunge ActionListener ai pulsanti e ListSelectionListener alle liste per gestire l'interazione. Chiama metodi del ChefService per popolare le viste ed eseguire azioni. Apre dialoghi modali (JDialog) per operazioni specifiche.

- UserMainFrame (JFrame): Dashboard dell'Utente. Strutturata per mostrare liste di corsi (disponibili e iscritti) e sessioni (online e in presenza). Utilizza JList con DefaultListModel. Permette iscrizione/disiscrizione ai corsi, conferma/annullamento partecipazione a sessioni pratiche, e visualizzazione dettagli sessioni (con link cliccabile per quelle online). Chiama metodi dell'UserService.

7.2 Dialoghi Modali (JDialog)

Queste finestre bloccano l'interazione con la finestra genitore finché non vengono chiuse. Sono usate principalmente dalla ChefMainFrame.

- GestioneSessioniDialog: Permette il CRUD completo per SessioneOnline e SessioneInPresenza relative a un Corso selezionato. Contiene JList e JButton. Utilizza JOptionPane.showConfirmDialog con pannelli personalizzati per l'input. Apre GestioneProgrammaDialog.

- GestioneIngredientiDialog: Permette di gestire la ComposizioneRicetta per una Ricetta selezionata. Contiene JList, JComboBox, JTextField, JButton. Apre JOptionPane per creare nuovi ingredienti.

- GestioneProgrammaDialog: Permette di gestire il ProgrammaSessione per una SessioneInPresenza selezionata. Contiene JList, JComboBox, JTextField, JButton.