TheKnife

Manuale Tecnico

Lorenzo Radice
Laurea Triennale in Informatica, Università degli Studi dell'Insubria
Matricola: 753252 — Sede Como

Anno Accademico 2024/2025

Indice

1 Progettazione

1.1 Use-Case Diagram

Per ciascuna tipologia di utente (ospite, cliente, ristoratore) sono stati individuati i casi d'uso principali, dalla ricerca di ristoranti alla gestione di recensioni e risposte. É stato redatto uno Use-Case Diagram che rappresentasse le principali interazioni tra gli utenti e il sistema.

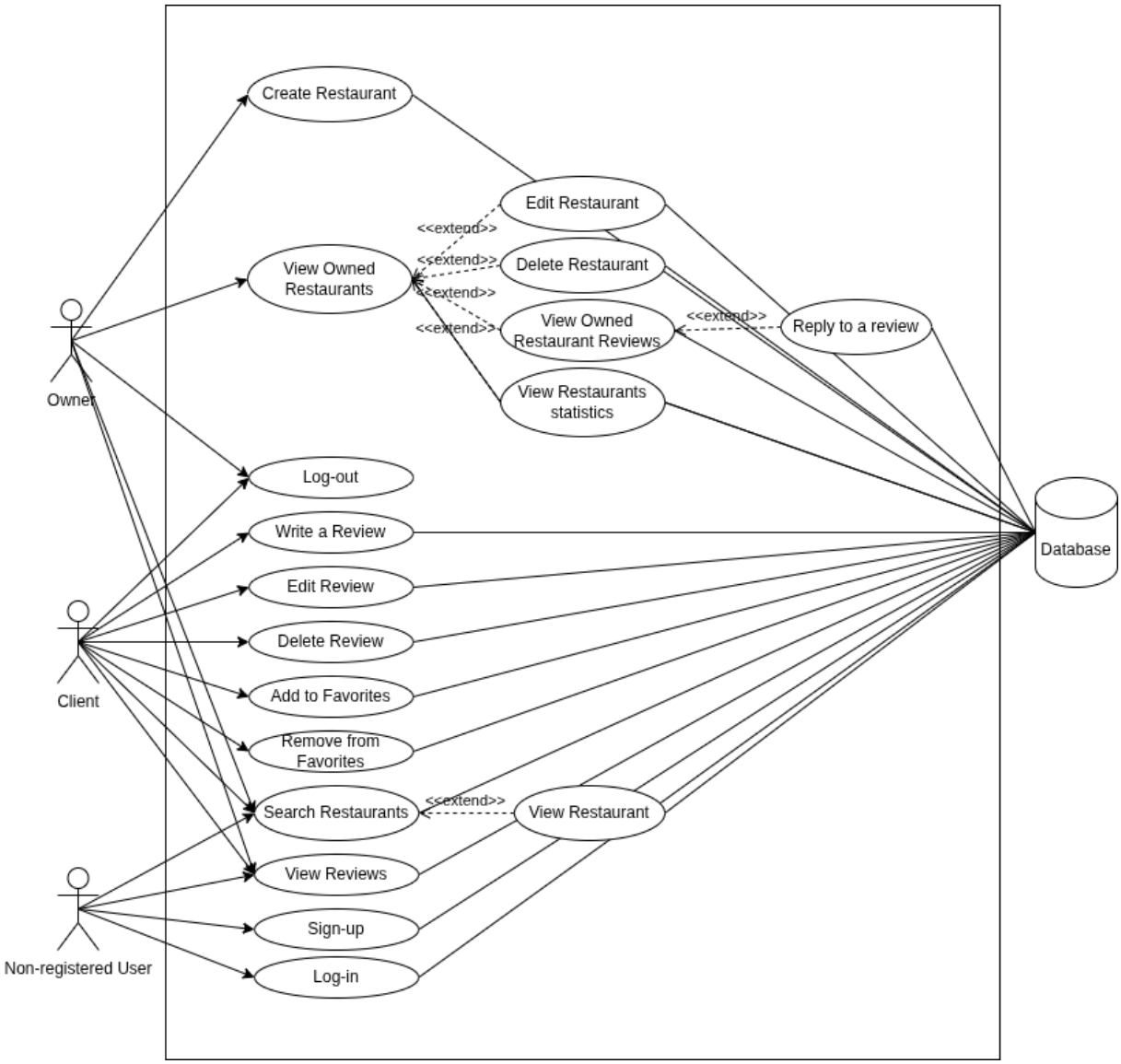

Figura 1: User-Case Diagram

Basandosi sul diagramma si è poi proceduto ad articolare le singole funzionalità in modo da poterle implementare in modo modulare attraverso la Progettazione Orientata agli Oggetti.

1.2 Class Diagram

Al fine di rappresentare le classi e le loro relazioni, dallo Use-Case Diagram è stato sviluppato un Class Diagram che potesse racchiudere le classi principali del sistema.

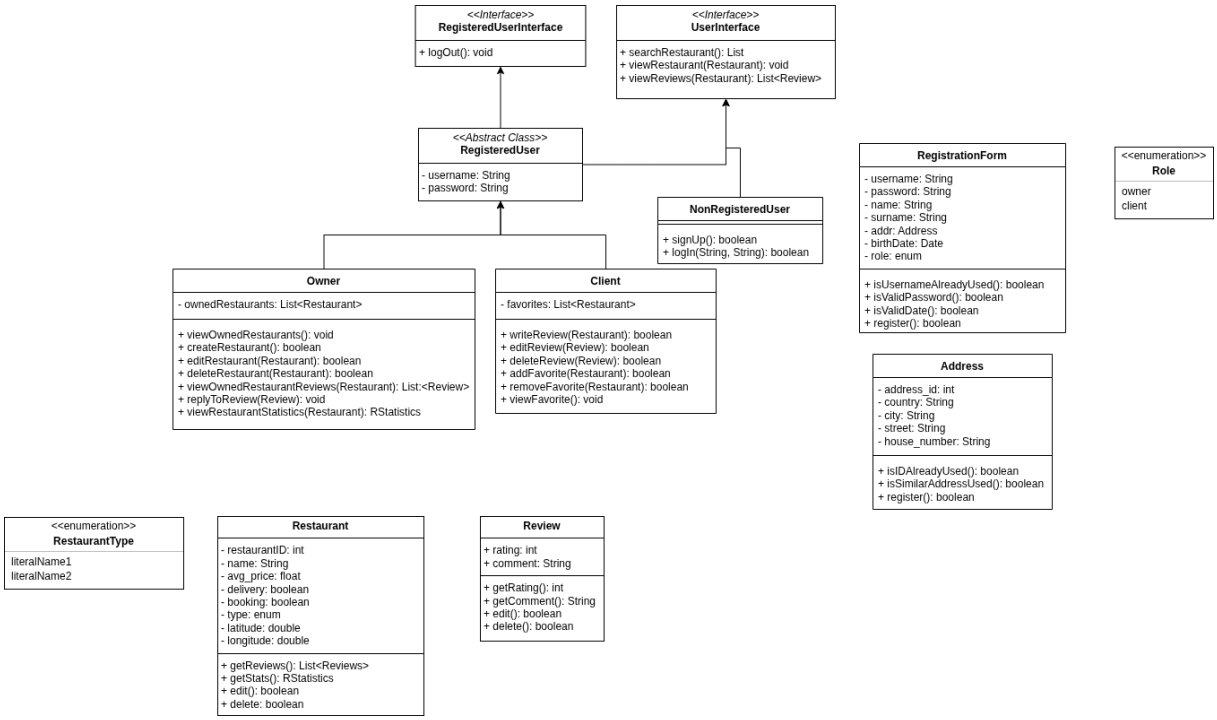

Figura 2: Class Diagram

Il diagramma è stato poi utilizzato per implementare le classi in Java, seguendo le relazioni e le associazioni indicate.

1.3 Architettura del Sistema

Il sistema è basato su un'architettura Client-Server.
Si è scelto di utilizzare Java RMI per la comunicazione tra client e server, in modo da poter sfruttare le funzionalità di Remote Method Invocation e garantire una comunicazione semplice ed efficace. Il server espone i servizi attraverso un Registry RMI sulla porta `1099`, permettendo ai client di accedere ai metodi remoti. Il client, a sua volta, si connette al server per invocare i metodi e ricevere le risposte.

É stato fondamentale definire delle interfacce comuni a Client e Server, in modo da garantire una comunicazione coerente e facilitare l'implementazione delle funzionalità. Per questa ragione il progetto è suddiviso in tre moduli principali:

- **client**: implementa il client e la sua interfaccia grafica
- **common**: contiene le interfacce comuni e le classi di utilità
- **server**: implementa il server e le funzionalità di gestione dei dati

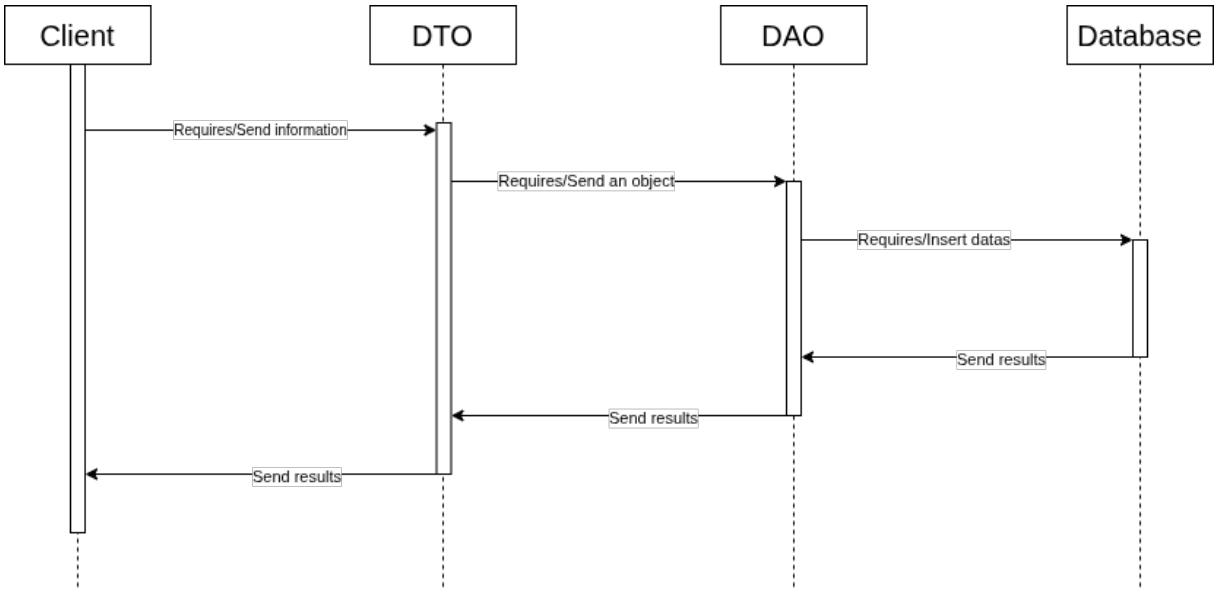

Figura 3: Interaction Diagram

2 Database

Si è scelto di utilizzare un database relazionale per la gestione dei dati dell'applicazione. Come database è stato scelto PostgreSQL.

2.1 Schema del Database

2.1.1 Diagramma Entità-Relazione

Il database è stato progettato realizzando un diagramma Entità-Relazione che racchiude i dati che si ritiene necessario gestire.

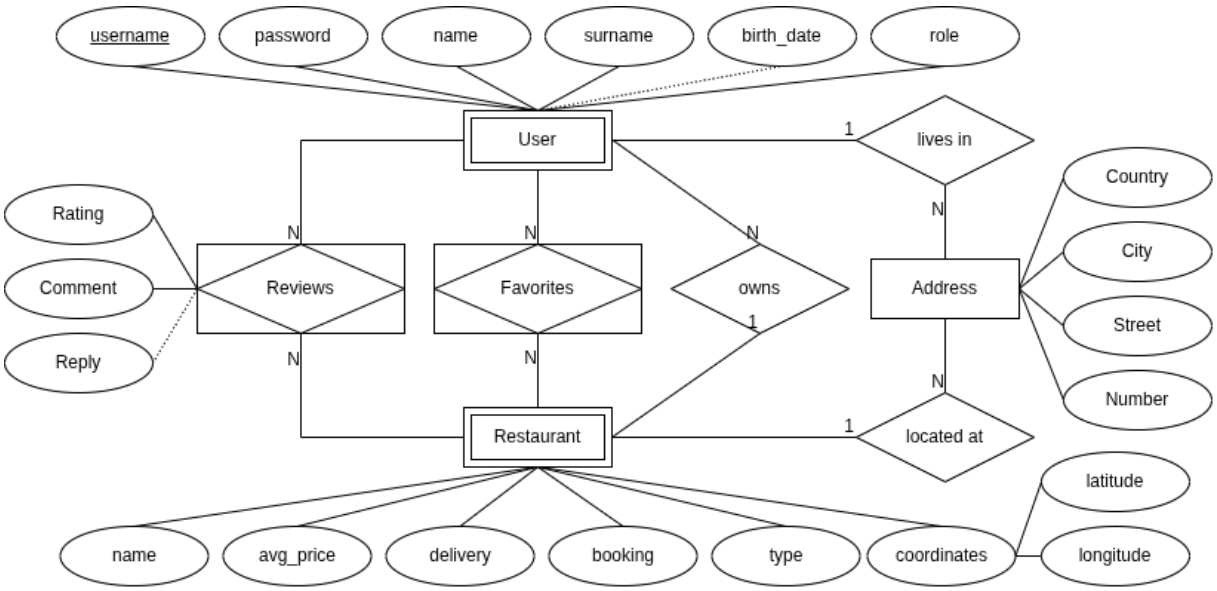

Figura 4: Diagramma ER

Come si evince dal diagramma l'entità principale è *Address* che contiene i dati relativi agli indirizzi. *User* e *Restaurant* sono in relazione con *Address* e a loro volta sono in relazione tra loro. *Review* e *Favorite* sono entità che collegano *User* e *Restaurant* per gestire le recensioni e i ristoranti preferiti dagli utenti.

2.1.2 Rielaborazione del Diagramma

Successivamente è stata eseguita una rielaborazione del diagramma per normalizzare le tabelle.

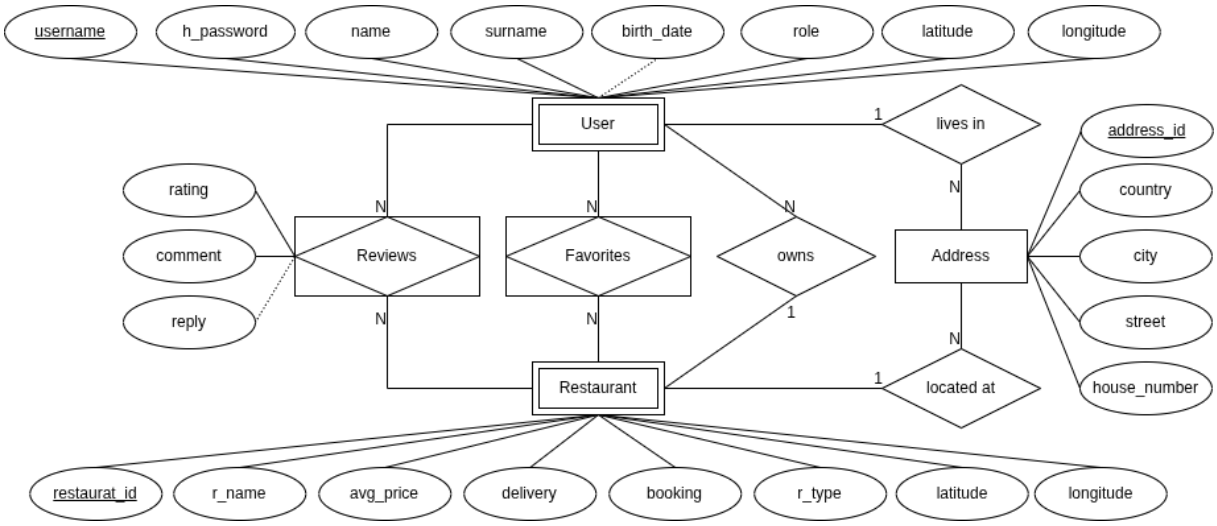

Figura 5: Diagramma ER Normalizzato

Dopo questa rielaborazione le coordinate da attributo derivato sono diventati attributi semplici. Tuttavia, nell'implementazione finale, si è deciso di spostare le coordinate nell'entità *Address* per una migliore gestione.

2.1.3 Schema

Infine si sono elencate le tabelle e i relativi attributi:

- **addresses** (address_id, country, city, street, house_number, latitude, longitude)
- **users** (username, h_password, name, surname, birth_date, role, address_id$^{\text{addresses}}$)
- **restaurants** (restaurant_id, r_owner, r_name, avg_price, delivery, booking, r_type, address_id$^{\text{addresses}}$)
- **favorites** (username$^{\text{users}}$, restaurant_id$^{\text{restaurants}}$)
- **reviews** (username$^{\text{users}}$, restaurant_id$^{\text{restaurants}}$, rating, comment, reply)

Questa lista delle tabelle è stata infine implementata nel database.

2.2 Tabelle

2.2.1 Addresses

La tabella *addresses* contiene gli indirizzi degli utenti e dei ristoranti.
Gli attributi sono:

- **address_id**: identificativo univoco dell'indirizzo
- **country**: nazione
- **city**: città
- **street**: via
- **house_number**: numero civico
- **latitude**: latitudine
- **longitude**: longitudine

Sono stati posti dei vincoli di integrità per garantire che le coordinate siano valide, nello specifico che la latitudine sia compresa tra -90 e 90 e la longitudine tra -180 e 180.

2.2.2 Users

La tabella *users* contiene gli utenti registrati nell'applicazione.
Gli attributi sono:

- **username**: identificativo univoco dell'utente
- **h_password**: password in hash
- **name**: nome
- **surname**: cognome
- **birth_date**: data di nascita
- **role**: ruolo dell'utente (owner o client)
- **address_id**: identificativo dell'indirizzo dell'utente

Il campo *role* è stato implementato creando un tipo di dato enumerativo che può assumere i valori *owner* o *client*.
Il campo *h_password* prevede che la password venga memorizzata in forma di hash per garantire la sicurezza. Si è scelto di utilizzare l'algoritmo di hashing Argon2 suggerito da OWASP.
Il campo *address_id* è una chiave esterna che fa riferimento alla tabella *addresses*. In caso di modifica dell'indirizzo di un utente, si aggiornerà il campo *address_id*, in caso di cancellazione dell'indirizzo non verrà cancellato l'utente.

2.2.3 Restaurants

La tabella *restaurants* contiene i ristoranti creati dai ristoratori.
Gli attributi sono:

- **restaurant_id**: identificativo univoco del ristorante
- **r_owner**: proprietario del ristorante (username dell'utente)
- **r_name**: nome del ristorante
- **avg_price**: prezzo medio del ristorante inserito dal ristoratore
- **delivery**: servizio di consegna disponibile (booleano)
- **booking**: servizio di prenotazione disponibile (booleano)
- **r_type**: tipo di cucina del ristorante (es. cinese, italiano, etc.)
- **address_id**: identificativo dell'indirizzo del ristorante

Il campo *r_owner* è una chiave esterna che fa riferimento alla tabella *users* e rappresenta il proprietario del ristorante. In caso di moifica di *username* dell'utente, si aggiornerà il campo *r_owner*, in caso di cancellazione dell'utente verrà cancellato anche il ristorante.
Il campo *address_id* è una chiave esterna che fa riferimento alla tabella *addresses*. In caso di modifica dell'indirizzo del ristorante, si aggiornerà il campo *address_id*, in caso di cancellazione dell'indirizzo non verrà cancellato il ristorante.
È stato posto un vincolo di integrità per garantire che il prezzo medio sia un numero positivo.

2.2.4 Favorites

La tabella *favorites* server a gestire i ristorante preferiti dagli utenti.
Gli attributi sono:

- **username**: identificativo dell'utente
- **restaurant_id**: identificativo del ristorante

Il campo *username* è una chiave esterna che fa riferimento alla tabella *users* e il campo *restaurant_id* è una chiave esterna che fa riferimento alla tabella *restaurants*. In caso di modifica dei due campi si aggiorneranno i rispettivi campi, in caso di cancellazione di un utente o di un ristorante, verrà cancellata la preferenza.

2.2.5 Reviews

La tabella *reviews* serve a gestire le recensioni che gli utenti possono lasciare ai ristoranti.
Gli attributi sono:

- **username**: identificativo dell'utente
- **restaurant_id**: identificativo del ristorante
- **rating**: valutazione da 1 a 5
- **comment**: commento della recensione
- **reply**: risposta del ristoratore alla recensione

Il campo *username* è una chiave esterna che fa riferimento alla tabella *users* e il campo *restaurant_id* è una chiave esterna che fa riferimento alla tabella *restaurants*. In caso di modifica dei due campi si aggiorneranno i rispettivi campi, in caso di cancellazione di un utente o di un ristorante, verrà cancellata la recensione.
Sul *rating* è stato posto un vincolo di integrità per garantire che sia un numero intero compreso tra 1 e 5.

3 Sitografia

- https://fxdocs.github.io/docs/html5/
- https://docs.oracle.com/javase/8/docs/technotes/guides/rmi/index.html
- https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html
- https://github.com/p-h-c/phc-winner-argon2

4 Server

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat