



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 IN INFROMATICA

INSEGNAMENTO DI BASI DI DATI E SISTEMI OPERATIVI I

ANNO ACCADEMICO 2019/2020

Documentazione

Applicativo Java

“Me Bell”

<Sistema Gestionale per Recensioni Turistiche>

A cura di:

Federico Gargiulo - N86002884

Antonio Garofalo - N86003129

Introduzione

Il progetto che abbiamo scelto tra le tracce disponibili è stato quello di un ***Sistema Gestionale per Recensioni Turistiche***, si chiedeva al gruppo di due studenti di creare una base di dati che venisse gestita da un applicativo java.

I vincoli imposti, sono stati prettamente 2:

-Un utente può unicamente fare una singola recensione per una determinata location.

-Le location devo essere gestite in 3 macro-specializzazioni che sarebbero:

- Alloggio
- Attrazione
- Ristorante

(N.B le specializzazioni devono essere ulteriormente raffinate)

E' stato pensato di fare una base di dati su Oracle (precisamente versione 11g facilmente configurabile) gestendo il tutto con l'applicativo "SQL developer".

La base di dati è stata modellata mediante Class Diagram con successiva revisione e ristrutturazione. Si noterà bene nella definizione di vincoli e tabelle la creazione anche di trigger che, per mancata presenza in Oracle, abbiamo inserito al fine di avere una base di dati dinamica che si aggiornasse automaticamente al cambiamento di specifici parametri e alcuni che velocizzassero gli inserimenti creando parallelamente altri inserimenti a catena da modificare con le informazioni desiderate.

L'applicativo JAVA è stato sviluppato su "Eclipse" con l'ausilio del plug-in "Windows builder".

L'applicativo inoltre è stato sottoposto ad un sistema di versionamento tramite repository offerte dal sito "GitHub" utilizzando appositi software per avere backup di tutto il progetto dal primo giorno di sviluppo.

Infine per i gruppi a 3 veniva chiesto un sistema di moderazione, tramite notifiche e controlli sulle recensioni fatte prima di essere pubblicate.

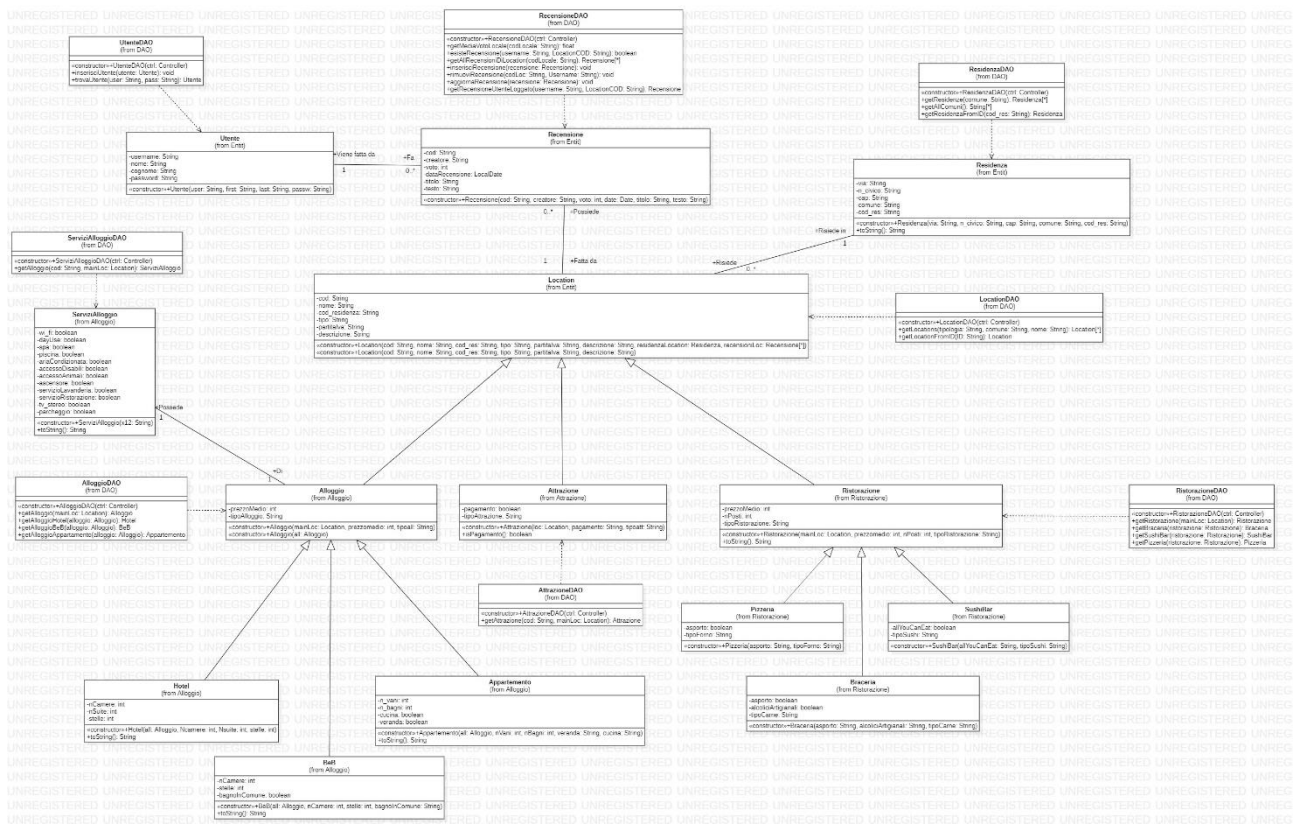
Class Diagram

Il class diagram rappresenta una visione completa dell'intero progetto, essendo molto voluminoso è stato redatto in due parti, la prima rispetta il problema posto con associazioni e specializzazioni insieme a tutte le classi DAO, la seconda parte invece è il "Controller insieme alla classe "DatabaseConnection".

Il problema è stato affrontato utilizzando il pattern DAO (Data Access Object) ovvero utilizzando classi apposite chiamate <nomeClasseEntità>DAO contenente tutti i metodi per la comunicazione con il DB.

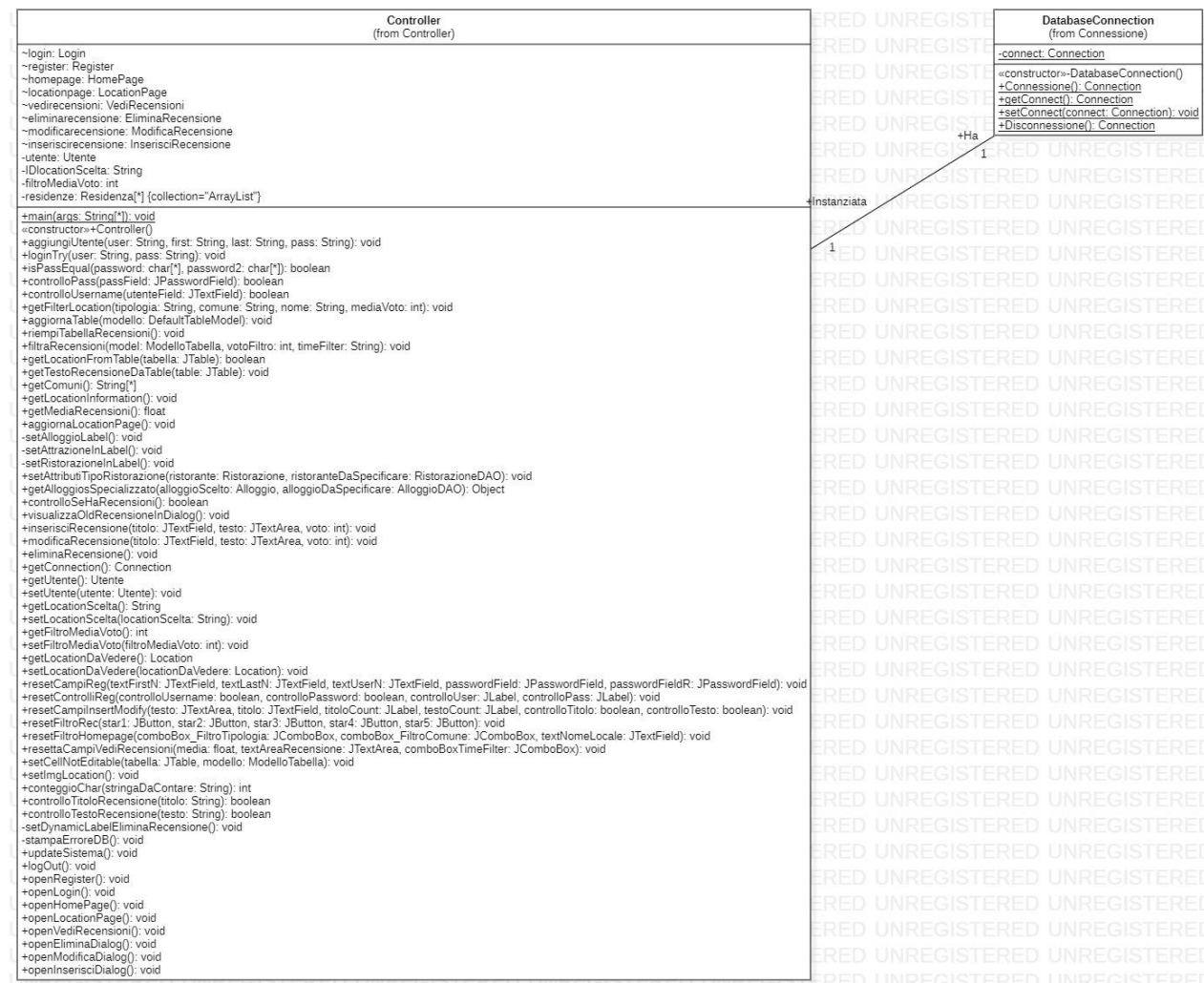
N.B

Per semplicità sono stati omessi getter e setter dato che rendevano voluminosamente ampie le classi.



Il class diagram sopra riportato rappresenta tutta la struttura utilizzata per sviluppare l'applicativo JAVA e il database che sarà poi sfruttato.

A seguire il controller:



Il controller ha un'unica connessione gestita dalla classe “DatabaseConnection” tramite il metodo `getConnection()` che controlla se c'è o no la connessione senza mai istanziarla.

La classe “DatabaseConnection” è sviluppata secondo il pattern “Singleton”.

«enumeration» Tipo Ristorazione	«enumeration» Tipologia Locale	«enumeration» Tipo Forno	«enumeration» Tipo Attrazione
Elenco completo Pizzeria Braceria SushiBar Pizza e Brace Pizza e Sushi Brace e Sushi	Alloggio Attrazione Ristorante	Entrambi A legna Elettrico	Chiesa Museo Parco Monumento Parco divertimenti Parco acquatico Centro sportivo Teatro Cinema Acquario Zona balneare Zoo
«enumeration» Tipo Alloggio	«enumeration» Tipo Sushi	«enumeration» Tipo Carne	
Hotel B&B Appartamento	Fusion Giapponese Cinese	Entrambe Importata Italiana	

“Tipi specificati”

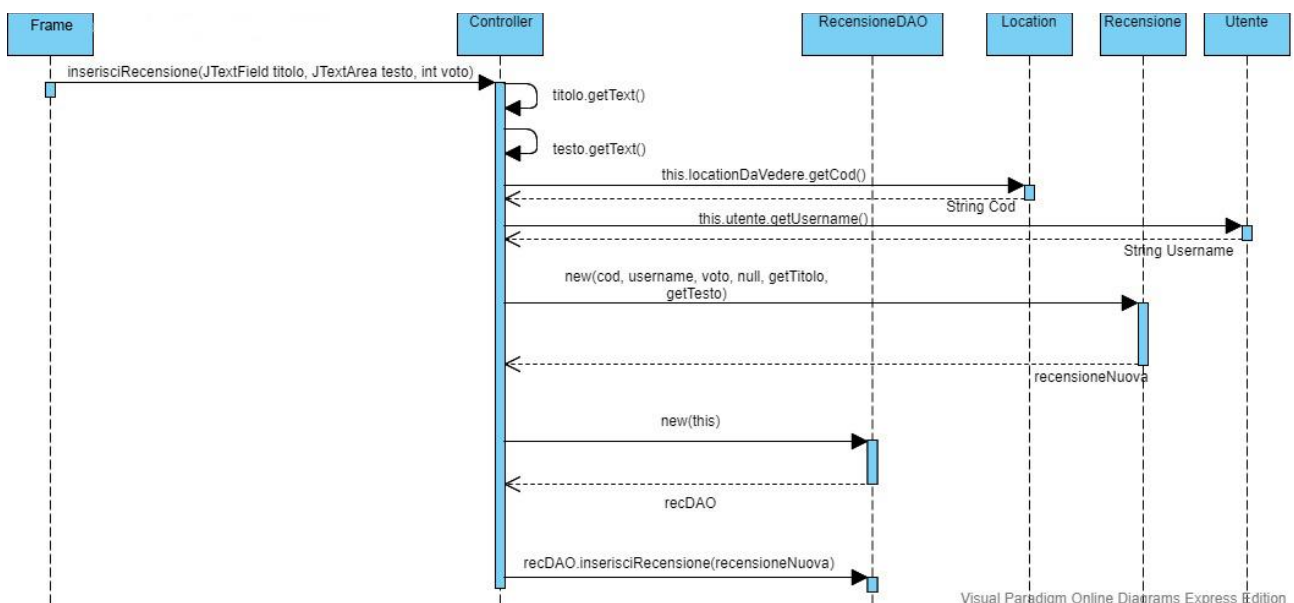
Sequence diagram

I sequence diagram che verranno riportati (in totale 5) rappresentano 5 funzioni più o meno complesse che sono state sviluppate per la soluzione del problema.

Le funzioni sono:

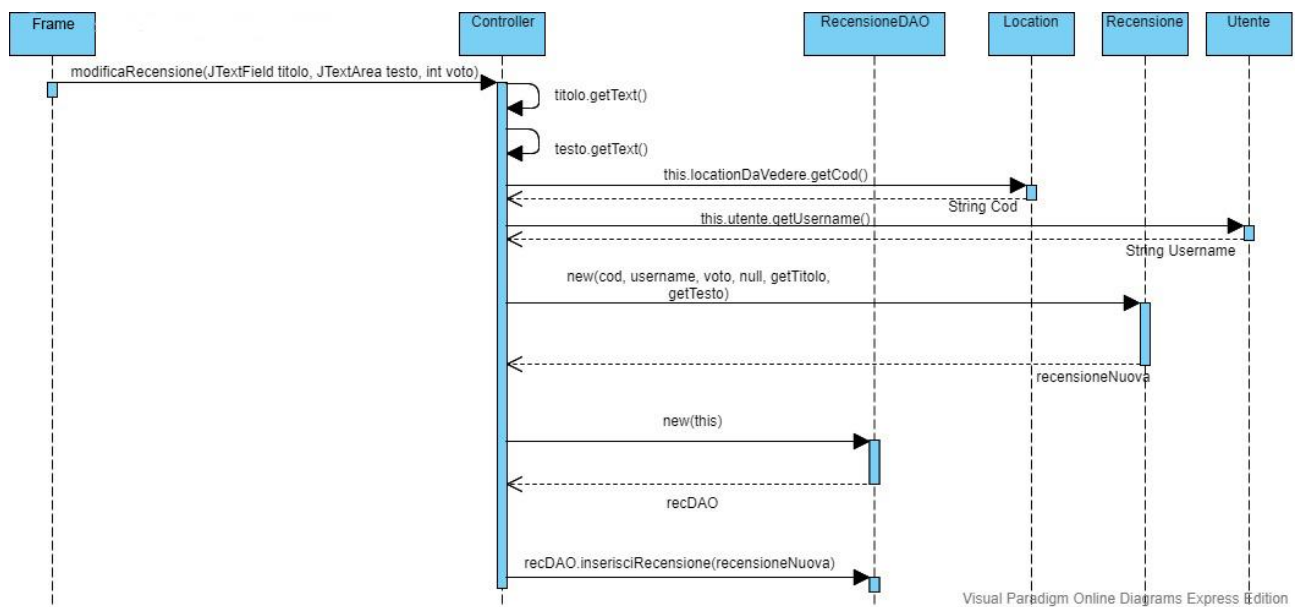
- Funzione di inserimento recensione;
- Funzione di modifica recensione;
- Funzione di eliminazione recensione;
- Funzione di login del programma;
- Funzione di aggiornamento della tabella della homepage del programma;

Funzione di inserimento:



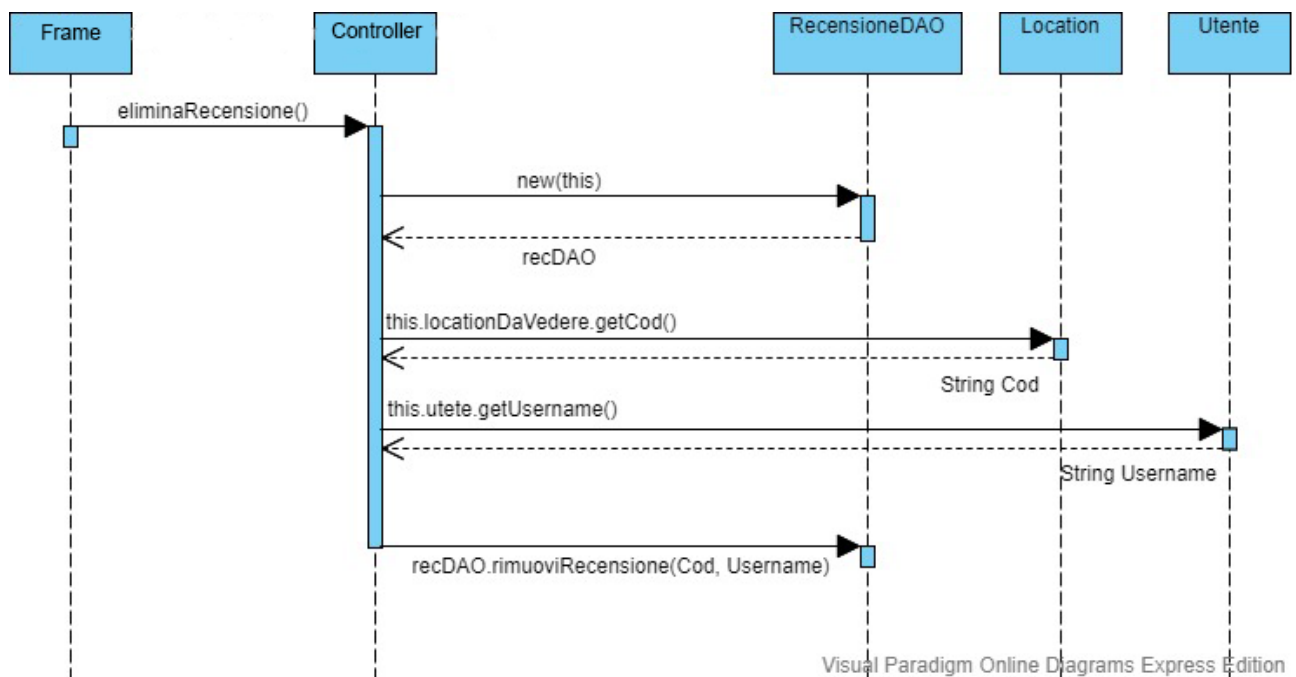
La funzione inserimento viene chiamata solo se l'utente ha accesso alla dialog di "Inserisci Recensione" (ovvero quando non ha ancora pubblicato una recensione per la location selezionata) e sfrutta il DAO di recensione per avviare un inserimento.

Funzione di modifica:



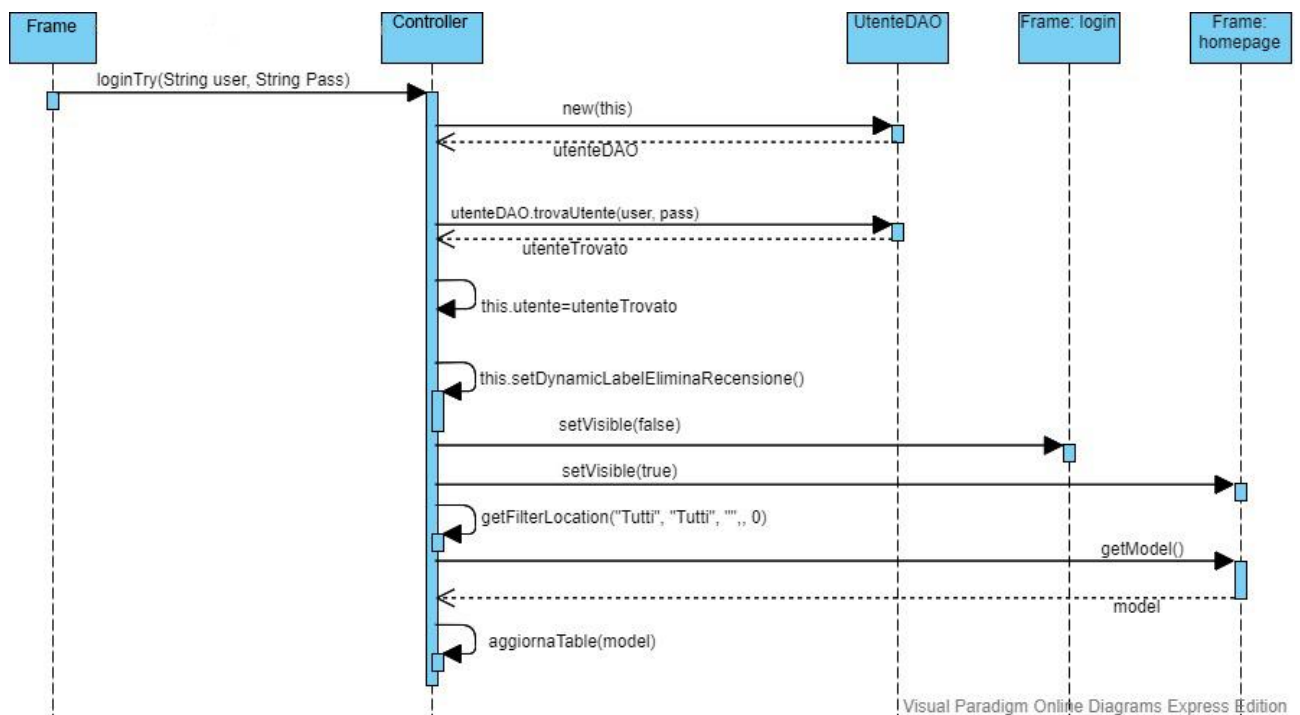
La funzione modifica viene chiamata solo se l'utente ha accesso alla dialog di "Modifica Recensione" (ovvero quando ha già pubblicato una recensione per la location selezionata) e sfrutta il DAO di recensione per avviare una modifica.

Funzione Eliminazione:



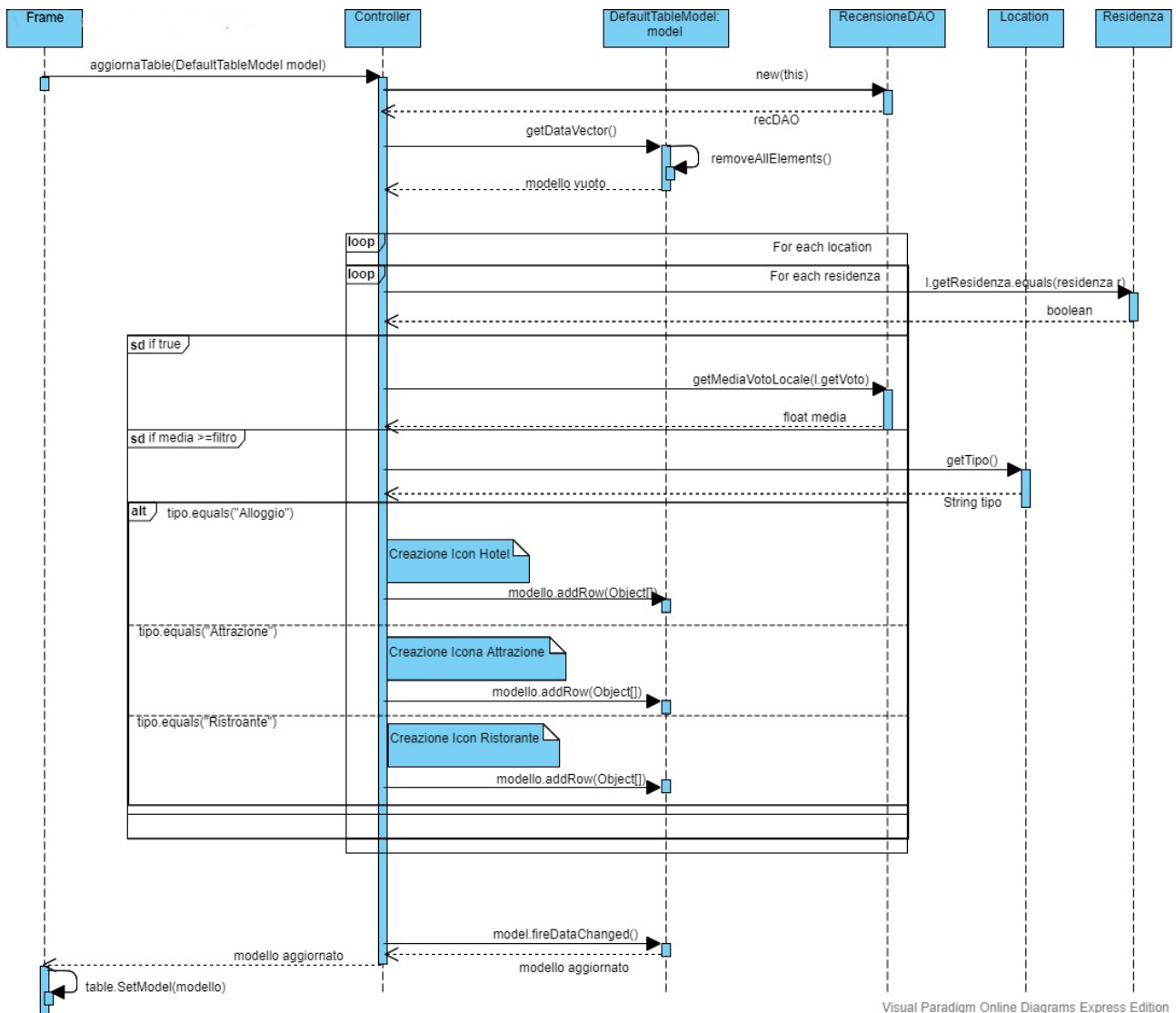
La funzione modifica viene chiamata solo se l'utente ha accesso alla dialog di "Elimina Recensione" (ovvero quando ha già pubblicato una recensione per la location selezionata) e sfrutta il DAO di recensione per avviare una eliminazione.

Funzione di login:



La funzione di login è chiamata dal frame iniziale "Login", tale frame (se i controlli di login sono "TRUE" dal tasto "Login" avvia la funzione che controlla se esiste un utente, se non esiste, il login è errato (lanciando l'eccezione).

Funzione di aggiornamento tabella in homepage:



La funzione descritta dal sequence diagram sopra riportato è una delle principali funzioni dell'applicativo, essa permette di visionare in formato tabellare tutto il database (o in parte tramite filtri appositi), per ovvie ragioni di voluminosità sono state omesse alcune funzioni getter e setter e la funzione di “Creazione” di una label con immagine che andrà a comparire in una cella apposita della tabella.

Mock-up

Nella prossima pagina è possibile visionare frame per frame il mock-up dell'intero programma insieme anche all'ausilio di link (posti su pulsanti) che lo rendono praticamente attendibile e accettabile come anteprima della versione DEMO che si andrà poi a discutere.