**TECNICO SVILUPPO SOFTWARE**

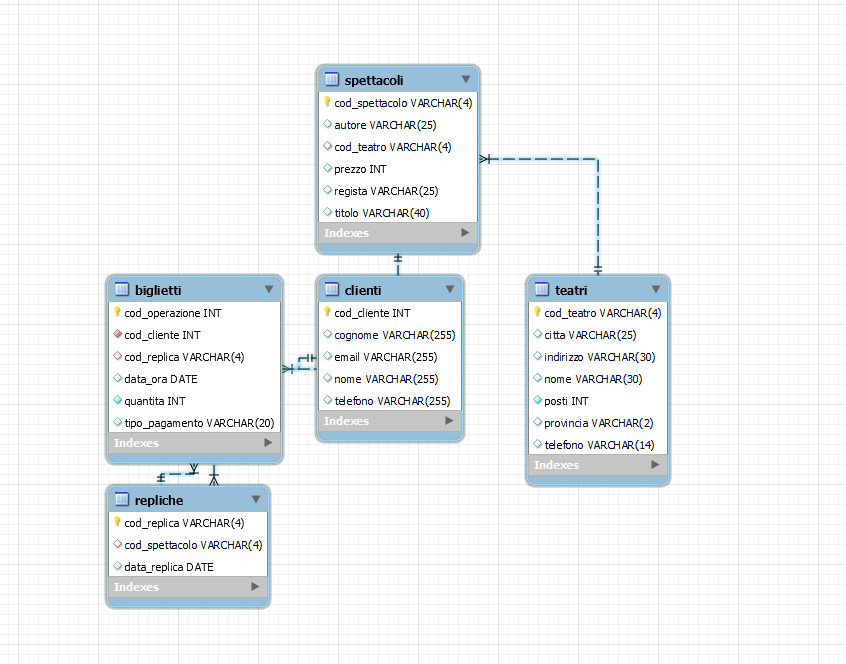
**RELAZIONE ESAME PROVA OPERATIVA**

**STEFANIA NERI**

**B22-104-2024**

Il progetto è stato realizzato utilizzando Java, Spring, Hibernate e MySQL per la gestione di backend e database, e Javascript, HTML, CSS e Bootstrap per la parte frontend.

Inizialmente è stato creato il database “teatro” in MySQL, all’interno del quale sono andata a generare le tabelle “spettacoli”, “biglietti”, “clienti”, “teatri” e “repliche”. Le tabelle sono state generate con Hibernate, tramite le entità che sono andata a creargli con Spring, andando a definire il tipo di dato per i vari attributi compresa anche la length nel caso di varchar. Inoltre sono state definite le relazioni tra le tabelle, così come esposto nel diagramma sottostante.



Successivamente ho caricato le query di insert per inserire i record nelle varie tabelle con i dati forniti dall’excel “TABELLE\_DATI”.

Avviando il progetto si atterra nella pagina di login chiamata “index”, ciò è stato possibile tramite il Model View Controller (MVC), con cui sono stati mappati i percorsi delle viste delle varie pagine che compongono il progetto. In questa prima pagina appare una schermata di accesso, in cui l’utente può inserire un Codice cliente, e premendo sul bottone “Accedi” si viene reindirizzati alla pagina “home”.

Nel caso in cui si tenti però di indicare un Codice cliente invalido, ossia non presente in database, apparirà un messaggio di errore e l’accesso non viene effettuato.

L’oggetto associato al Codice cliente inserito viene salvato in Local Storage per poter avere accesso ai dati nelle pagine successive, questo ci servirà quando si andrà a prenotare un biglietto e si dovrà sapere chi effettivamente sta prenotando.

Nella barra di navigazione sarà inoltre presente nelle pagine successive il bottone “Logout”, con cui viene svuotata la Local Storage e l’utente verrà riportato alla pagina di accesso.

In questa seconda pagina “home” appare la lista degli spettacoli prenotabili, filtrati in JavaScript in modo da visualizzare per ogni spettacolo il teatro in cui verrà riprodotto, in che giorno, e altri dettagli reperiti da database. Per avere accesso a questi dati ho fatto tre fetch per le tabelle “spettacoli”, “repliche” e “teatro”, utilizzando le API che ho creato nel backend nel REST Controller, in cui sono andata a mappare gli endpoint per gestire ogni tabella necessaria, in questo caso con l’annotation GetMapping per fare una richiesta GET e visualizzare i dati.

Per ogni spettacolo è inoltre possibile scegliere la modalità di pagamento e la quantità dei biglietti che si vogliono prenotare per la replica di quel determinato spettacolo. Premendo sul bottone “Prenota” verrà eseguita una fetch con il metodo “POST” che andrà ad aggiungere nella tabella “biglietti”, richiamando il PostMapping inserito nel REST. Inoltre verrà in concomitanza eseguita una fetch con il metodo “PATCH”, richiamando il PatchMapping del REST, per andare a ridurre i posti disponibili ancora liberi nel teatro prenotato.

Questa pagina di elenco spettacoli è stata realizzata andando a creare, per ogni record presente in tabella, una card con i propri elementi HTML, generati interamente in JavaScript in modo tale da poter avere tante card quanti sono i record in tabella, e nessuna card in caso di tabella vuota.

L’input “Quantità” è stato configurato in modo da non accettare numeri negativi. Inoltre, se si tenta di inserire un numero di biglietti maggiori rispetto ai posti ancora disponibili in quel determinato teatro, la POST non verrà eseguita e apparirà un messaggio che comunica che si sta prenotando più posti rispetto a quelli consentiti. Questo controllo è stato fatto impostando una condizione che, prima di eseguire la POST, confronti l’importo della quantità inserita con il valore dei posti nella tabella del teatro.

Al click su “Prenota”, nel caso in cui la POST vada a buon fine, apparirà sotto al bottone un messaggio dinamico con un countdown che anticipa i secondi mancanti al reindirizzamento alla pagina successiva, ovvero “prenotazioni”.

Dopo aver prenotato un biglietto infatti, si atterra nella pagina che espone una tabella con la lista dei biglietti acquistati. Allo stesso modo in cui sono state create le card nella pagina precedente, anche qui ho creato gli elementi delle righe e colonne in JavaScript, per far apparire per ogni record nella tabella “biglietti” una riga con i dati di ognuno. Per ogni biglietto, e quindi per ogni riga esposta in questa tabella, verrà visualizzato:

* Nome e cognome di chi ha acquistato il biglietto
* Tipo di pagamento
* Numero dei biglietti prenotati
* Data in cui è stata fatto l’acquisto del biglietto
* Titolo dello spettacolo
* Autore
* Regista
* Teatro

01/08/2024

Stefania Neri