

Prenotazione Intelligente di Classi Universitarie in Sicurezza

PICUS

Documento di Progettazione

1 Architettura generale

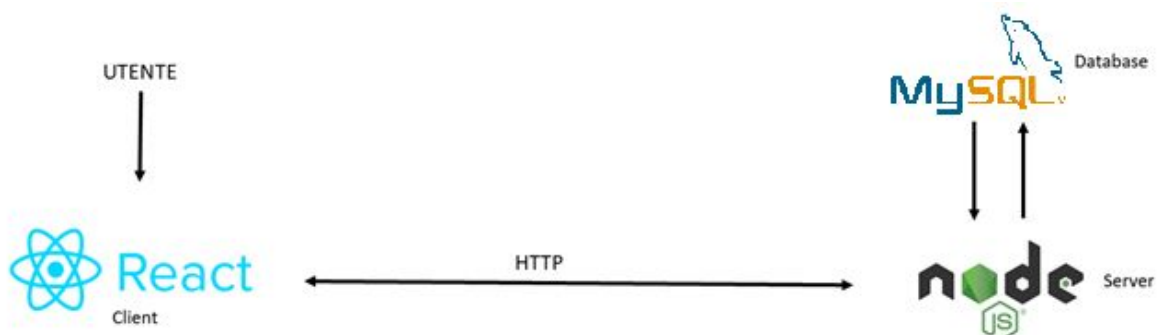
Quella mostrata di seguito, è l'intera architettura generale:



dove:

- **Il Client:** Mette a disposizione un'interfaccia grafica che permette, in base all'utente, di eseguire le funzionalità descritte nel documento di analisi.
- **Il Database:** Il database offre funzionalità di storage di dati.
- **Il server:** Il server preleva le richieste del cliente attraverso l'interfaccia grafica e elabora tale richieste -> l'elaborazione consiste nel prelievo dei dati presenti sul database.

2 Architettura logica

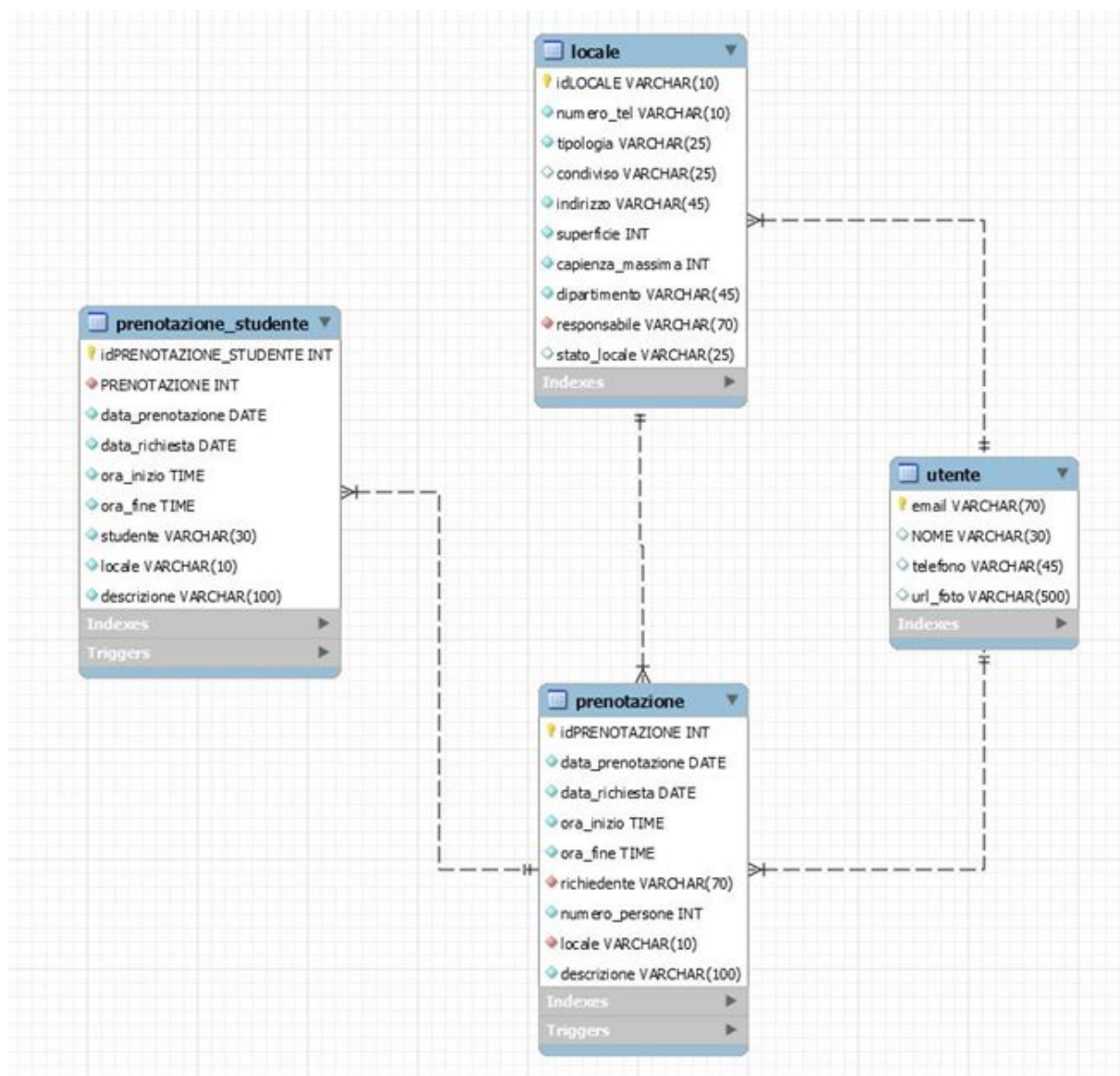


Dove:

- **Il server:** La tecnologia supportata dal server è NodeJS. Il server espone delle API RESTful che gestiscono la tipologia di richiesta; In base alla tipologia (GET,PUT,POST, etc), quest'ultima verrà elaborata opportunamente. Poiché sul canale viaggiano dati sensibili (querystring, url), tale canale dovrà essere criptato utilizzando HTTPS (HTTP over TLS).

- **Il Client:** L'interfaccia grafica per il client è realizzata interamente in React. Tale tecnologia offre una serie di tool e funzionalità che permettono all'utente di interagire col sistema in maniera smart e semplice. Il framework, tra le altre cose, garantisce sicurezza per i tipi di attacchi XSS e SQLInjection.
- **Il Database:** Il database utilizzato è di tipo relazionale: MySQL. Il database permette, inoltre, connessioni basate su TLS; ciò significa che il database potrebbe essere presente sia localmente sul server sia in remoto senza esporre il sistema a problemi di sicurezza.

3 Modello entità-relazione



Il Modello Entità-Relazioni è composto da quattro tabelle:

- **Locale**
- **Utente**
- **Prenotazione**
- **Prenotazione_studente**

La tabella **Locale** contiene gli attributi che sono stati descritti nel documento di analisi nella funzionalità “Inserimento Locali”. Si può osservare la presenza di alcune aggiunte:

1. **idLocale**: primary key auto-incrementante;
2. **stato_locale**: un locale può essere “aperto” o “chiuso” -> un locale può essere prenotato correttamente da un docente solo se il locale si trova nello stato “aperto”. Un locale che viene inserito nel sistema ha di default lo stato “chiuso”. Quest’ultimo può essere modificato, successivamente, da un operatore nella gestione dei locali.

La tabella **Utente** contiene tutti i docenti che si sono registrati alla piattaforma. In particolare vengono memorizzati: email, nome, telefono ed url_foto (campo utilizzato per mostrare eventuali foto del docente)

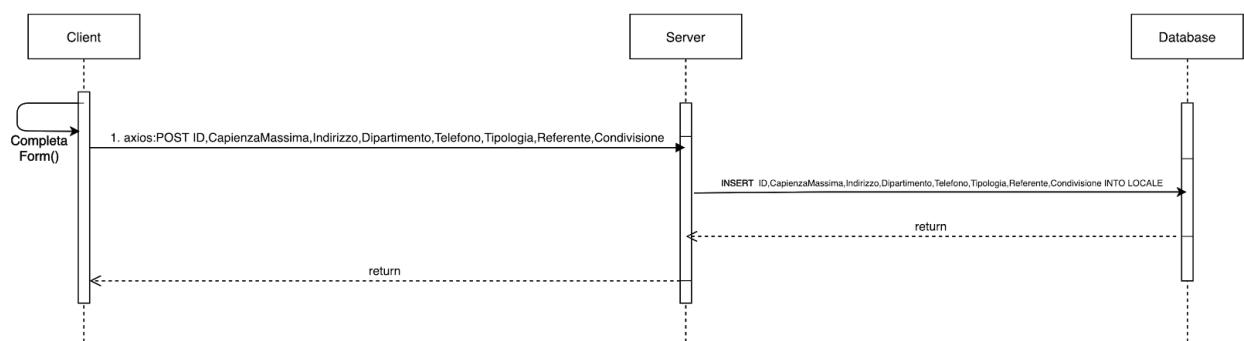
La tabella **Prenotazione** memorizza tutte le prenotazioni effettuate dai vari docenti: all’interno di questa tabella ci sono due foreign key:

1. Richiedente (ovvero il docente che sta effettuando la prenotazione del locale) è un attributo referenziato al campo email della tabella utente (controlla appunto che ci sta docente con quella email nella tabella utente).
2. Locale è un attributo referenziato al campo idLocale presente nella tabella Locale (stesso ragionamento di prima sul campo richiedente).

La tabella **Prenotazione_studente**, a differenza della precedente, contiene tutte le prenotazioni dei vari studenti: molti campi (data_richiesta, ora_inizio, ora_fine) sono ripresi dalla tabella Prenotazione, mentre altri, come il campo Prenotazione (primary key autoincrementante) è referenziato al campo idPrenotazione dell’altra tabella.

4 Dinamiche di interazione

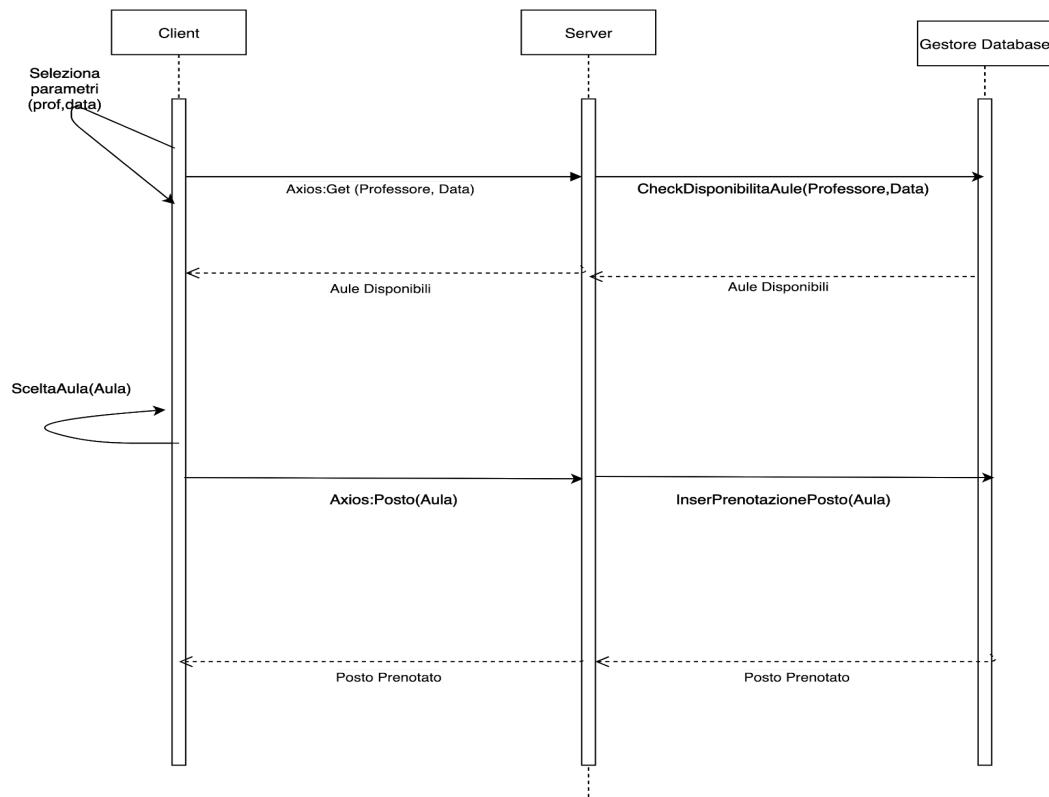
Sequence diagram *Inserimento Locale*



L’utente, in questo caso l’operatore, interfacciandosi con l’interfaccia client, completa le form inserendo i dati del locale da inserire. Dopo aver fatto ciò e dopo aver inviato la richiesta al

server, quest'ultimo contatta il database il quale inserisce una nuova tupla all'interno dell'apposita tabella.

Sequence diagram *Prenota posto in locale*



L'utente, in questo caso lo studente, prima di prenotare un posto nell'aula prenotata da un docente, richiede a Picus, completando delle apposite form, l'elenco preciso delle aule che il docente di riferimento ha prenotato. Dopo aver completato le form e inviato la richiesta al server, quest'ultimo interroga il database per ricercare le aule che corrispondono ai dati immessi. Una volta mostrate le aule allo studente, quest'ultimo può scegliere a quale lezione prenotarsi cliccando un apposito bottone che scatenerà l'inserimento della nuova prenotazione nell'apposita tabella nel database.