

DOCUMENT DE PROIECTARE ARHITECTURALA – SISTEM BOOKING HOTELIER

PARTEA I – PROIECTAREA ARHITECTURALĂ

1. Introducere

1.1 Scopul sistemului

Sistemul permite gestionarea rezervărilor hoteliere pentru un lanț cu locații multiple, oferind interfețe dedicate pentru clienți, angajați și administratori.

1.2 Definiții și acronime

- **Client:** Utilizator care rezervă camere, plătește și lasă recenzii.
- **Administrator:** Gestionează hotelurile, camerele și utilizatorii.
- **Contabil:** Gestionează veniturile și comisioanele.
- **Rezervare:** Confirmarea unei camere într-un anumit hotel, pentru o anumită perioadă.

1.3 Documente referință

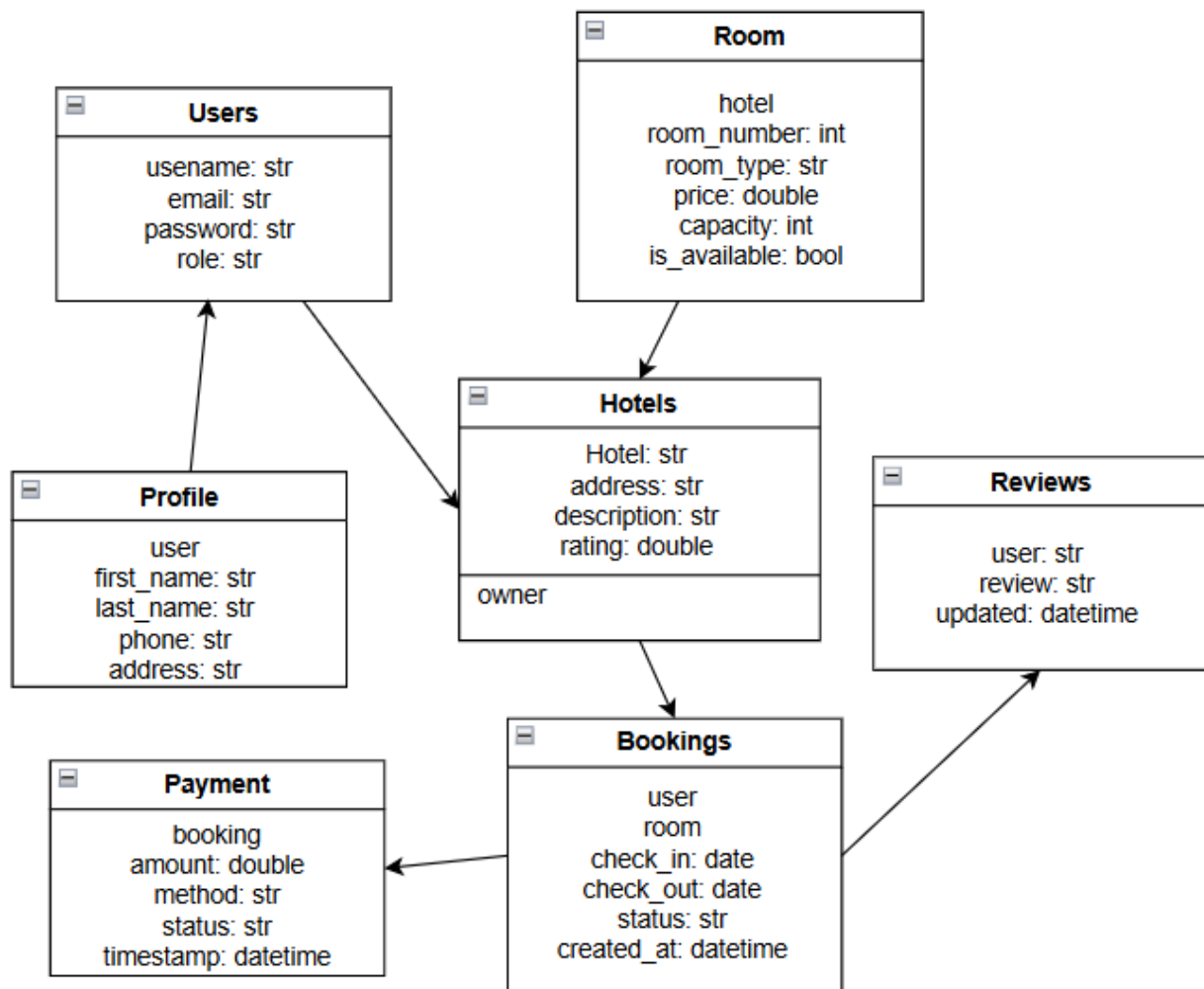
- „Specificația cerințelor” – descrie cerințele funcționale și nefuncționale
- Modele similare: Booking, Airbnb

2. Obiective de proiectare

- Separarea clară a subsistemelor pentru client, angajat și administrator.
- Interfață intuitivă, responsive.
- Sistem modular, ușor de extins.
- Securitate a datelor utilizatorilor.
- Răspuns rapid și scalabilitate.

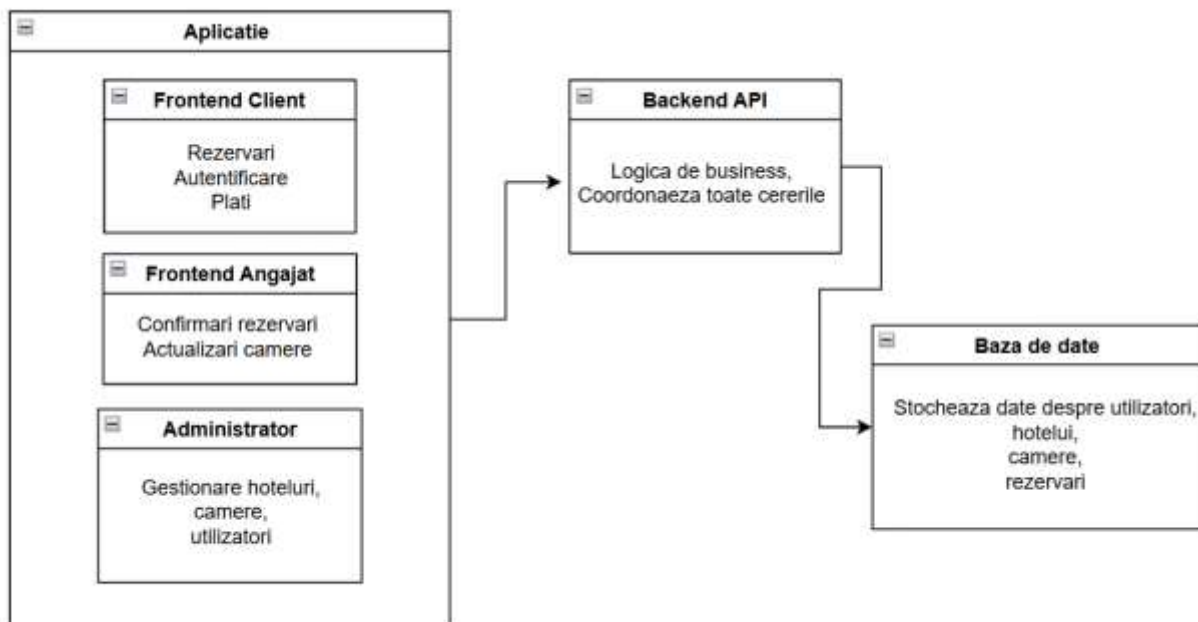
3. Arhitectura propusă

3.1 Decompoziția în subsisteme și diagrame UML



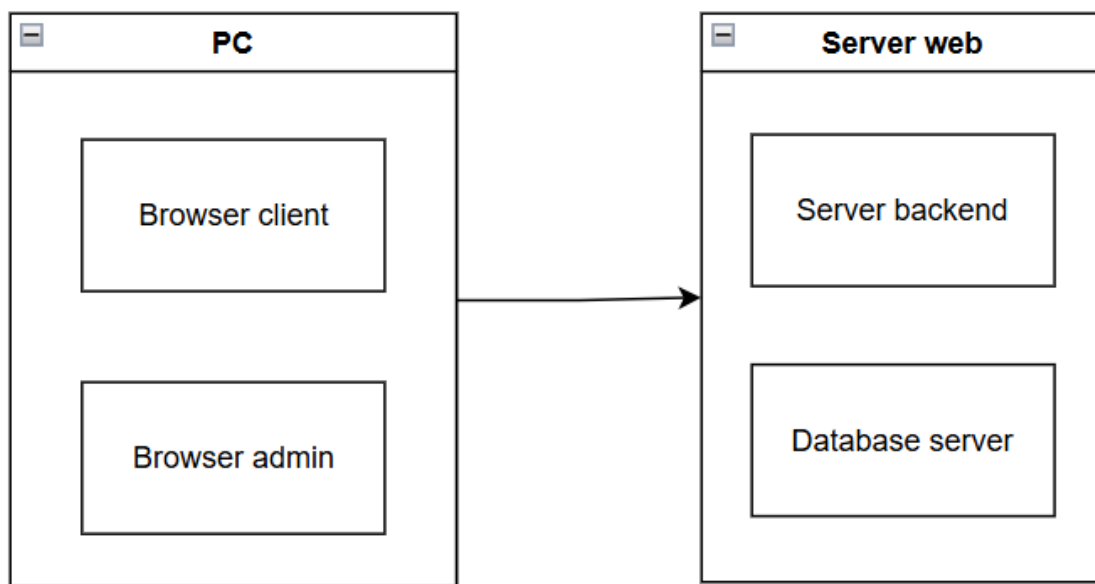
Subsistemele propuse:

- **Frontend Client:** permite înregistrarea, rezervarea, plata și recenziile.
- **Frontend Angajat:** permite vizualizarea rezervărilor, confirmarea lor și actualizarea camerelor.
- **Frontend Admin:** adaugă hoteluri, camere, gestionează utilizatori.
- **Backend API:** logică de business, conectarea la baza de date.
- **Baza de date:** stochează toate informațiile (utilizatori, rezervări, camere, hoteluri).



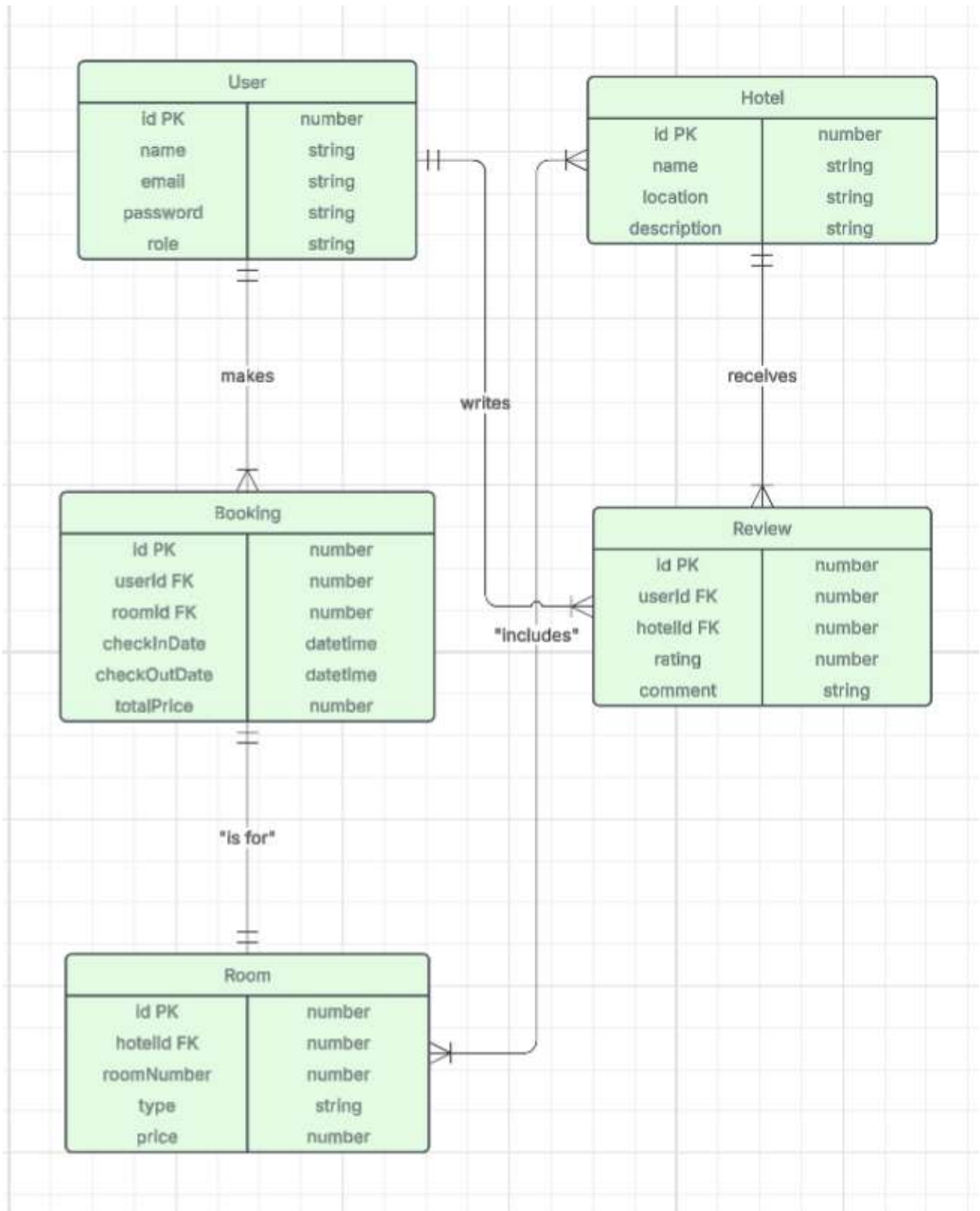
3.2 Distribuția pe platforme

- **Frontend Web** – Next.js
- **Backend** – Python Django
- **DB** – SQL



3.3 Managementul datelor persistente

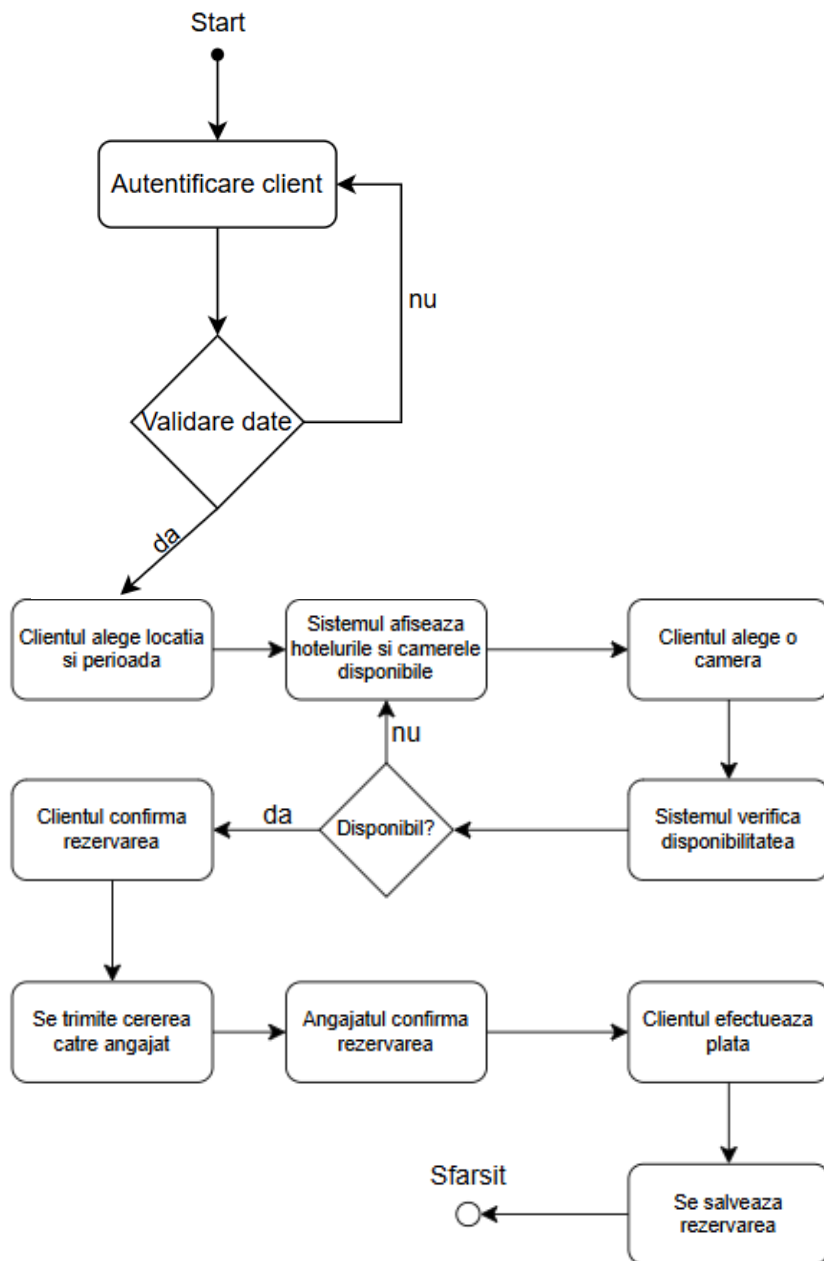
- Se folosește o bază de date relațională.
- Tabele esențiale: Utilizatori, Hoteluri, Camere, Rezervări, Recenzii.



3.4 Controlul accesului utilizatorilor

- Clienți, angajați și administratori au roluri diferite.
- Acces restricționat la interfețele respective.

3.5 Fluxul global al controlului



3.6 Tratarea condițiilor limită

- La pornirea sistemului: verifică conexiunea la baza de date.
- În caz de eroare (ex. cameră deja rezervată): trimite mesaj de eroare și opțiuni alternative.

- Oprește sistem: se salvează datele active și se face log de backup.

PARTEA II – PLANIFICAREA IMPLEMENTĂRII

Echipa și roluri generale

Membru	Rol General	Responsabilități
Ramona	Frontend Client	Login, rezervări, interfață UI
David	Frontend Admin	Panou hoteluri, camere, utilizatori
Stefan	Backend Developer	API rezervări, camere, autentificare
Andrei	Database & Testare	Proiectare BD, testare funcțională

Iterația 1 – Sarcini detaliate

Creare formular login/înregistrare client

Ramona Vizualizare hoteluri

Selectare cameră și rezervare

Dashboard admin (lista hoteluri, adăugare cameră)

David Modificare stare cameră (rezervată/liberă)

Stefan Endpoint-uri REST: login, createUser, getHotels, createBooking

Creare model conceptual DB în SQL

Andrei Populare DB cu date test

Scriere test pentru endpoint rezervare