Book hotel application

1.1, Select the city from the list.

1.2, Select room type.

1.3, Choose the date in calendar.

1.4, (Optional) Allow to book a hotel for any number of days.

2, Choose the hotel from the list of found matches.

3, Type own name, password and confirm own choice with button.

Notes:

1. There are N (≈4) cities in the list.

2. Each city has at least M (≈5) hotels.

3. There are at least R (≈10) rooms in each hotel (each room belongs to one of possible types). 4. Reservation can be done for the next 14 days

5. The system should identify the situation when two users are trying to book the last remaining slot simultaneously and prevent overbooking (the system should print the message “this day is no longer available” and let the user select another day).

6. The user must select some day from the list; it is not possible to reserve a time slot for “any available day”.

User interface

1. After the desired city, date, and the room type are chosen, the app displays the list of matching hotels, having available rooms for the specified criteria. If no hotels are found, the system suggests to the user to change search parameters.

2. After confirming the booking, the system displays the details (date, city, hotel name, room type).

3. Registration requires entering a unique combination of login/password.

4. If a certain user is logged in, there is no need to ask for a username anymore.

User Interface

グラフィカル ユーザー インターフェイス

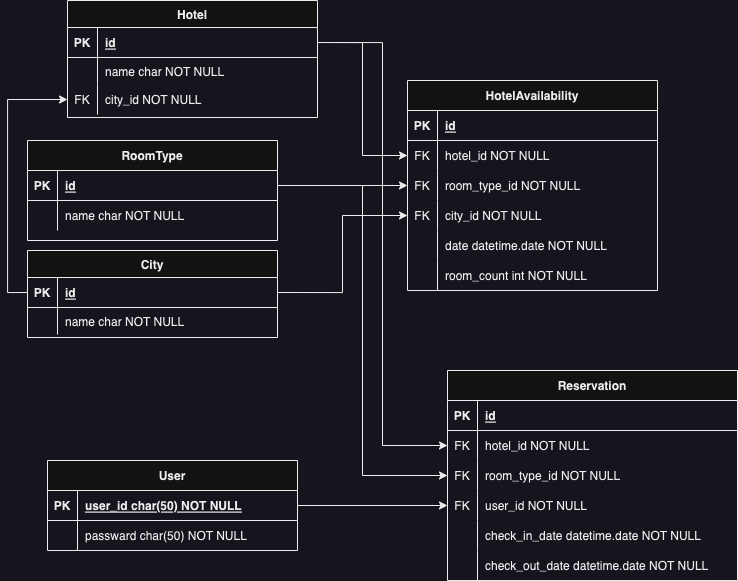
自動的に生成された説明

Proposed Architecture

グラフ, ツリーマップ図

自動的に生成された説明

Database Structure



**Django Hotel Booking Application**

This is a simple Django hotel booking application that allows users to search for hotels, view available rooms, and make reservations. Below is a brief explanation of each functionality in the application:

**Functionality**

1. **Top Page (topPage)**:
   * This is the landing page of the application.
   * Users are redirected to this page when they log out from the system.
   * It provides a welcoming interface with options to explore further.
2. **Search Page (searchPage)**:
   * This page allows users to search for hotels based on certain criteria.
   * It displays a form where users can select a city, room type, check-in date, and check-out date.
   * The form includes a date picker to ensure proper date selection.
   * The available search options include cities, room types, and dates within the next two weeks from the current date.
3. **Hotel List (hotelList)**:
   * This page displays a list of hotels and their available rooms based on the search criteria from the search page.
   * When the user submits the search form, it queries the database for available hotels and rooms that match the specified criteria.
   * The system checks the availability of rooms for consecutive dates to ensure users can book multiple nights if needed.
   * Hotels with zero available rooms are not displayed in the list.
   * Error and success messages from the reservation process are shown on this page.
4. **Reservation (reservation)**:
   * This page allows users to make a reservation for a specific hotel and room type.
   * Users can select the check-in and check-out dates for their reservation.
   * The system checks the availability of rooms for the selected dates to ensure a valid reservation.
   * If the reservation is successful, the room count for the selected dates is reduced by one.
   * If the room count becomes zero for any date during the selected stay, the reservation fails.
   * Users must be logged in to make a reservation, but if not logged in, they are prompted to create a new user account.
5. **Login (login\_user)**:
   * This page provides a form for users to log into the system.
   * Users need to provide their username and password to log in.
   * If the login is successful, the user is redirected to the top page.
   * If the login fails, an appropriate error message is displayed.
6. **Logout (logout\_user)**:
   * This function logs out the currently logged-in user and redirects them to the top page.
7. **My Page (my\_page)**:
   * This page displays the reservations made by the logged-in user.
   * If the user is not logged in, they are redirected to the login page.

**Models**

The application uses the following Django models:

* City: Represents a city where hotels are located.
* RoomType: Represents types of rooms available in hotels (e.g., single, double, suite).
* Hotel: Represents a hotel with its name, location, etc.
* HotelAvailability: Represents the availability of rooms in hotels for specific dates.
* Reservation: Represents a user's reservation with details like the hotel, room type, check-in, and check-out dates.