

Hotel Booking Management

Funzionalità

Descrizione

Il progetto è composto da un applicativo WEB rivolto ai gestori di attività alberghiere.

È suddiviso in due sezioni:

1. La prima sezione è rivolta agli utenti finali i quali potranno effettuare una prenotazione presso la struttura alberghiera: potranno selezionare il periodo di permanenza, il numero di ospiti, recapiti ed eventuali note. In una seconda fase è possibile selezionare una o più camere in base alla disponibilità corrente e inoltre dovranno essere inseriti i dettagli personali di chi effettua la prenotazione.

Le prenotazioni effettuate saranno soggette a conferma da parte della struttura.

2. La seconda sezione è rivolta agli amministratori della struttura alberghiera che potranno inserire, aggiornare, eliminare e confermare le prenotazioni relative ai giorni seguenti alla data odierna. È prevista la possibilità di inserimento, aggiornamento ed eliminazione sia delle stanze a disposizione e delle utenze all'interno del sistema.

Esiste anche una tipologia d'utenza detta "Viewer" che avrà accesso ai dati di amministrazione sopracitati in modalità di sola lettura.

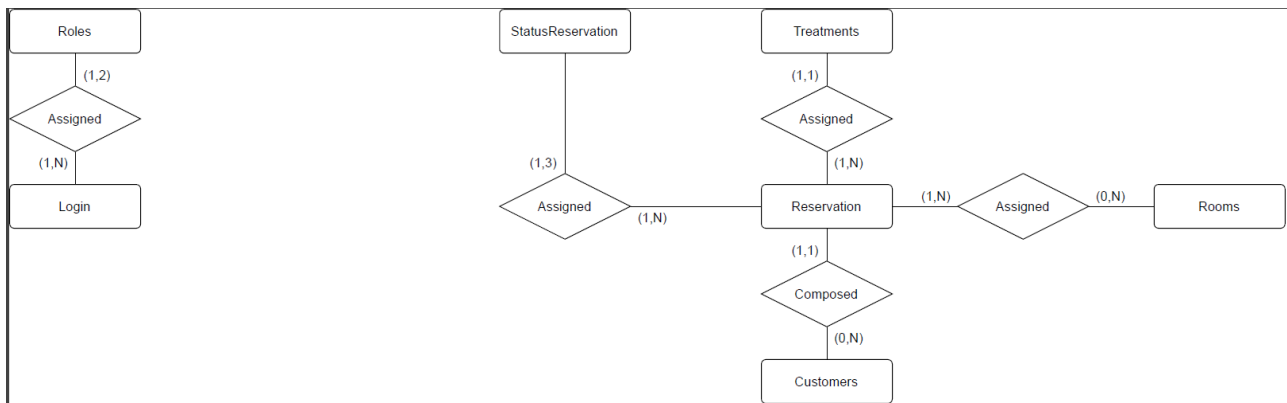
Tecnologie

L'applicativo è stato sviluppato con due tipologie di tecnologie:

- MySQL
- PHP: Viene impiegato il framework Laravel che garantisce la creazione di un'applicazione WEB applicando i costrutti e le regole della programmazione orientata agli oggetti sia nella parte UI sia nella parte di integrazione con il DB. Il pattern di sviluppo è di tipo MVC.

Diagrammi

Diagramma Entità-Relazione



Schema Logico

LOGICAL SCHEMA

Roles(idrole, descriptionrole)

Login(idlogin, name, surname, username, password)

StatusReservation(idstatusreservation, descriptionbookingstatus)

Treatments(idtreatment, descriptiontreatment)

Rooms(idroom, roomname, roomcapacity, roomcostpernight)

Customers(idcustomer, customername, customersurname, customergender, customerbirthdate, customerfiscalcode, customerbirthplace, customercity, customerprovince, customercitizenship, customerdocumentType, customerdocumentnumber, customerdocumentplaceofissue, customerdocumentcityofissue, customerdocumentprovinceofissue)

Reservation(idreservation, checkin, checkout, contacts, treatment, bookingstatus, adultsnumber, kidsnumber, newbornnumber, editabledata, notes)

Booking(idbooking, reservation, room)

Query

Qui di seguito saranno elencate le query utilizzate per il corretto funzionamento dell'applicazione raggruppate per tabella (Controller) di riferimento.

Per la creazione delle query è stato utilizzato (per una migliore integrazione con il framework) la funzione denominata "Query Builder" di Laravel il cui funzionamento è descritto in questa pagina (<https://laravel.com/docs/8.x/queries>).

Le query riportate sono da considerarsi "traduzioni" dalle funzioni di Laravel a SQL nativo.

Query riferite alla tabella Booking

Metodo INDEX:

```
SELECT *  
FROM Booking  
WHERE delete_at IS NULL
```

Metodo SHOW:

```
SELECT *  
FROM Booking  
WHERE idbooking = {{IDBooking}} AND deleted_at IS NULL
```

Metodo SHOWBYRESERVATION:

```
SELECT IDReservation  
FROM Booking booking INNER JOIN Rooms rooms on booking.room =  
rooms.idroom  
WHERE reservation = {{IDReservation}} AND booking.deleted_at IS NULL
```

Metodo STORE:

```
INSERT INTO Booking (idbooking,reservation,room)
VALUES ({{IDBooking}}, {{Reservation}}, {{Room}})
```

Metodo UPDATE:

```
UPDATE Booking
SET reservation={{Reservation}}, room={{Room}}
WHERE IDbooking = {{IDBooking}};
```

Metodo DESTROY:

```
DELETE FROM Booking
WHERE idbooking={{IDBooking}}
```

Query riferite alla tabella Customers

Metodo INDEX:

```
SELECT *
FROM Customer
WHERE delete_at IS NULL
```

Metodo SHOW:

```
SELECT *
FROM Customer
WHERE idcustomer = {{IDcustomer}} AND deleted_at IS NULL
```

Metodo SHOWBYRESERVATION:

```
SELECT *
FROM Customer
WHERE reservation = {{IDReservation}} AND deleted_at IS NULL
```

Metodo STORE:

```
INSERT INTO Customer (idcustomer, customername, customersurname,  
customergender, customerbirthdate, customerfiscalcode,  
customerbirthplace, customerbirthcity, customerbirthprovince,  
customercitizenship, customerdocumentType,  
customerdocumentnumber, customerdocumentplaceofissue,  
customerdocumentcityofissue, customerdocumentprovinceofissue,  
reservation)
```

```
VALUES ({{idcustomer}}, {{customername}}, {{customersurname}},  
{{customergender}}, {{customerbirthdate}}, {{customerfiscalcode}},  
{{customerbirthplace}}, {{customerbirthcity}},  
{{customerbirthprovince}}, {{customercitizenship}},  
{{customerdocumentType}}, {{customerdocumentnumber}},  
{{customerdocumentplaceofissue}}, {{customerdocumentcityofissue}},  
{{customerdocumentprovinceofissue}}, {{reservation}})
```

Metodo UPDATE:

```
UPDATE Customer
```

```
SET customername={{customername}},  
customersurname={{customersurname}},  
customergender={{customergender}},  
customerbirthdate={{customerbirthdate}},  
customerfiscalcode={{customerfiscalcode}},  
customerbirthplace={{customerbirthplace}},  
customerbirthcity={{customerbirthcity}},  
customerbirthprovince={{customerbirthprovince}},  
customercitizenship={{customercitizenship}},  
customerdocumentType={{customerdocumentType}},  
customerdocumentnumber={{customerdocumentnumber}},  
customerdocumentplaceofissue={{customerdocumentplaceofissue}},  
customerdocumentcityofissue={{customerdocumentcityofissue}},  
customerdocumentprovinceofissue={{customerdocumentprovinceofiss  
ue}}, reservation={{reservation}}
```

```
WHERE idcustomer={{idcustomer}};
```

Metodo DESTROY:

```
DELETE FROM Customer  
WHERE Idcustomer=1
```

Query riferite alla tabella Login:

Metodo INDEX:

```
SELECT *  
FROM Login  
WHERE delete_at IS NULL
```

Metodo SHOW:

```
SELECT *  
FROM Login  
WHERE idlogin = {{IDLogin}} AND deleted_at IS NULL
```

Metodo STORE:

```
INSERT INTO Login (idlogin, name, surname, username, password, role)  
VALUES ({{idlogin}}, {{name}}, {{surname}}, {{username}}, {{role}})
```

Metodo UPDATE:

```
UPDATE Login  
SET idlogin={{idlogin}}, name={{name}}, surname={{surname}},  
username={{username}}, role={{role}}  
WHERE IDLogin = {{IDLogin}};
```

Metodo DESTROY :

```
DELETE FROM Login  
WHERE Idlogin=1
```

Query riferite alla tabella Reservation

Metodo INDEX:

```
SELECT *  
  
FROM Reservation R INNER JOIN Treatments T ON T.idtreament =  
R.treatment INNER JOIN StatusReservation S ON S.idstatusreservation =  
R.bookingstatus  
  
WHERE R.delete_at IS NULL
```

Metodo SHOW:

```
SELECT *  
  
FROM Reservation  
  
WHERE idreservation = {{IDreservation}} AND deleted_at IS NULL
```

Metodo STORE:

```
INSERT INTO Reservation (idreservation, checkin, checkout, contacts,  
treatment, bookingstatus, adultsnumber, kidsnumber, newbornnumber,  
editabledata, notes)  
  
VALUES ({{idreservation}}, {{checkin}}, {{checkout}}, {{contacts}},  
{{treatment}}, {{bookingstatus}}, {{adultsnumber}}, {{kidsnumber}},  
{{newbornnumber}}, {{editabledata}}, {{notes}})
```

Metodo UPDATE:

```
UPDATE Reservation  
  
SET idreservation={{idreservation}}, checkin={{checkin}},  
checkout={{checkout}}, contacts={{contacts}}, treatment={{treatment}},  
bookingstatus={{bookingstatus}}, adultsnumber={{adultsnumber}},  
kidsnumber={{kidsnumber}}, newbornnumber={{newbornnumber}},  
editabledata={{editabledata}}, notes={{notes}}  
  
WHERE IDreservation=1;
```

Metodo DESTROY:

```
DELETE FROM Reservation  
WHERE Idreservation=1
```

Query riferite alla tabella Role

Metodo INDEX:

```
SELECT *  
FROM Role  
WHERE delete_at IS NULL
```

Metodo SHOW:

```
SELECT *  
FROM Role  
WHERE idrole={{idrole}} AND deleted_at IS NULL
```

Metodo STORE:

```
INSERT INTO Role (idrole, descriptionrole)  
VALUES ({{idrole}}, {{descriptionrole}})
```

Metodo UPDATE:

```
UPDATE Role  
SET descriptionrole={{descriptionrole}}  
WHERE idrole = {{idrole}};
```

Metodo DESTROY:

```
DELETE FROM Role  
WHERE idrole={{idrole}}
```


Query riferite alla tabella Room

Metodo INDEX:

```
SELECT *  
FROM Room  
WHERE delete_at IS NULL
```

Metodo INDEXAVAILABLEROOMS:

```
SELECT DISTINCT room.*  
FROM rooms room  
WHERE room.idroom NOT IN (  
    SELECT room2.idroom  
    FROM rooms room2 INNER JOIN booking book ON book.room =  
    room2.idroom  
    INNER JOIN reservations reservation ON book.reservation =  
    reservation.idreservation  
    WHERE (reservation.checkin BETWEEN ? AND ?) OR  
    (reservation.checkout BETWEEN {{CheckInDate}} AND  
    {{CheckOutDate}})  
)
```

Metodo SHOW:

```
SELECT *  
FROM Room  
WHERE idroom = {{IDRoom}} AND deleted_at IS NULL
```

Metodo STORE:

```
INSERT INTO Booking (idroom, roomname, roomcapacity,  
roomcostpernigth)  
  
VALUES ({{idroom}}, {{roomname}}, {{roomcapacity}},  
{{roomcostpernigth}})
```

Metodo UPDATE:

```
UPDATE Room  
  
SET idroom={{idroom}}, roomname={{roomname}},  
roomcapacity={{roomcapacity}},  
roomcostpernigth={{roomcostpernigth}}  
  
WHERE IDroom = {{idroom}};
```

Metodo DESTROY :

```
DELETE FROM Room  
  
WHERE IDroom={{IDroom}}
```

Query riferite alla tabella StatusReservation

Metodo INDEX:

```
SELECT *  
  
FROM StatusReservation  
  
WHERE delete_at IS NULL
```

Metodo SHOW:

```
SELECT *  
  
FROM StatusReservation  
  
WHERE idstatusreservation={{IDstatusreservation}} AND deleted_at IS  
NULL
```

Metodo STORE:

```
INSERT INTO StatusReservation (idstatusreservation,  
descriptionbookingstatus)  
VALUES ({{idstatusreservation}}, {{descriptionbookingstatus}})
```

Metodo UPDATE:

```
UPDATE StatusReservation  
SET descriptionbookingstatus={{descriptionbookingstatus}}  
WHERE IDstatusreservation = {{idstatusreservation}};
```

Metodo DESTROY :

```
DELETE FROM StatusReservation  
WHERE IDstatusreservation = {{idstatusreservation}};
```

Query riferite alla tabella Treatment

Metodo INDEX:

```
SELECT *  
FROM Treatment  
WHERE delete_at IS NULL
```

Metodo SHOW:

```
SELECT *  
FROM Treatment  
WHERE idtreatment = {{IDtreatment}} AND deleted_at IS NULL
```

Metodo STORE:

```
INSERT INTO Treatment (idtreatment, descriptiontreatment)
VALUES ({{idtreatment}}, {{descriptiontreatment}})
```

Metodo UPDATE:

```
UPDATE Treatment
SET descriptiontreatment={{descriptiontreatment}}
WHERE IDtreatment={{idtreatment}};
```

Metodo DESTROY:

```
DELETE FROM Treatment
WHERE Idtreatment={{idtreatment}}
```