# Hotels

Progetto di tecnologie web

Dumitru Frunza 135801

# Requisiti

Il progetto segue la traccia numero uno proposta dal docente:

“Applicazione Web per la gestione di un sistema di prenotazioni alberghiere in stile Booking.com.
L'applicazione è pensata per essere usata da utenti anonimi ed utenti registrati:
● gli utenti anonimi possono cercare alloggi disponibili in base al periodo di tempo e alle caratteristiche desiderate; nel momento in cui desiderano effettuare la prenotazione attraverso la piattaforma devono registrarsi alla stessa
● gli utenti registrati possono effettuare prenotazioni e gestirle (modifica e cancellazione), e vederne lo storico
● i proprietari delle strutture possono, previa registrazione al sito, inserire le caratteristiche dell'albergo, comprensive di tipologia di struttura, servizi offerti, numero e tipo di camere disponibili, prezzi degli alloggi, locazione geografica, foto, ...
Il sistema deve consentire la ricerca delle strutture alberghiere in base alla disponibilità in un determinato periodo ed alla presenza o meno di determinate caratteristiche (es, presenza di piscina, spa, ristorante): le strutture trovate in seguito alla ricerca devono essere mostrate in ordine decrescente di prezzo complessivo per la durata dell'alloggio. Il sistema deve gestire la disponibilità nelle strutture decrementando all'atto di ogni prenotazione per il periodo indicato. Deve inoltre dare agli utenti registrati la possibilità di modificare/eliminare le prenotazioni fatte precedentemente, modificando di conseguenza le prenotazioni. Un utente registrato può anche mettersi in lista d'attesa per una struttura già occupata in un determinato periodo e ricevere una notifica nel momento in cui la struttura divenisse disponibile in base a cancellazioni di prenotazioni.”

# Database e classi di modelli

È stato come database sqlite su consiglio del professore. Di seguito le tabelle usate:

## User

È default di django, un utente può avere n Reservation oppure n WaitLine

## Hotel

La tabella contiene tutte le informazioni utili per descrivere un alberg In particolare free_time è un campo utilizzato per creare oggetti di tipo Activities. Ogni attività dentro free_time è distinta da uno spazio e nel caso l'attività sia comprese da più di una parola è possibile scrivere utilizzando gli underscore: ping pong diventa ping_pong. Questa scelta implementativa permette di inserire qualsiasi tipo di attività ma si corre il rischio di errori da parte dell'utente.

## Rooms

Ogni Hotel può avere n Rooms. In questo Modello si salvano informazioni utili per la camera e permette di nominare una camera, nel caso in cui fosse bisogno di distinguere ad esempio per piani.

## Cost

Ogni stanza ha un oggetto di tipo Cost, che descrive prezzo per periodo. L'obiettivo era quello di avere un pulsante "Aggiungi periodo" che aggiungeva una nuova riga tramite script javascript, permettendo la variazione di prezzo in base alla stagione. Questa opzione è stata messa da parte perché creava seri problemi nel caso in cui i periodi dovessero essere aggiornati.

## Activities

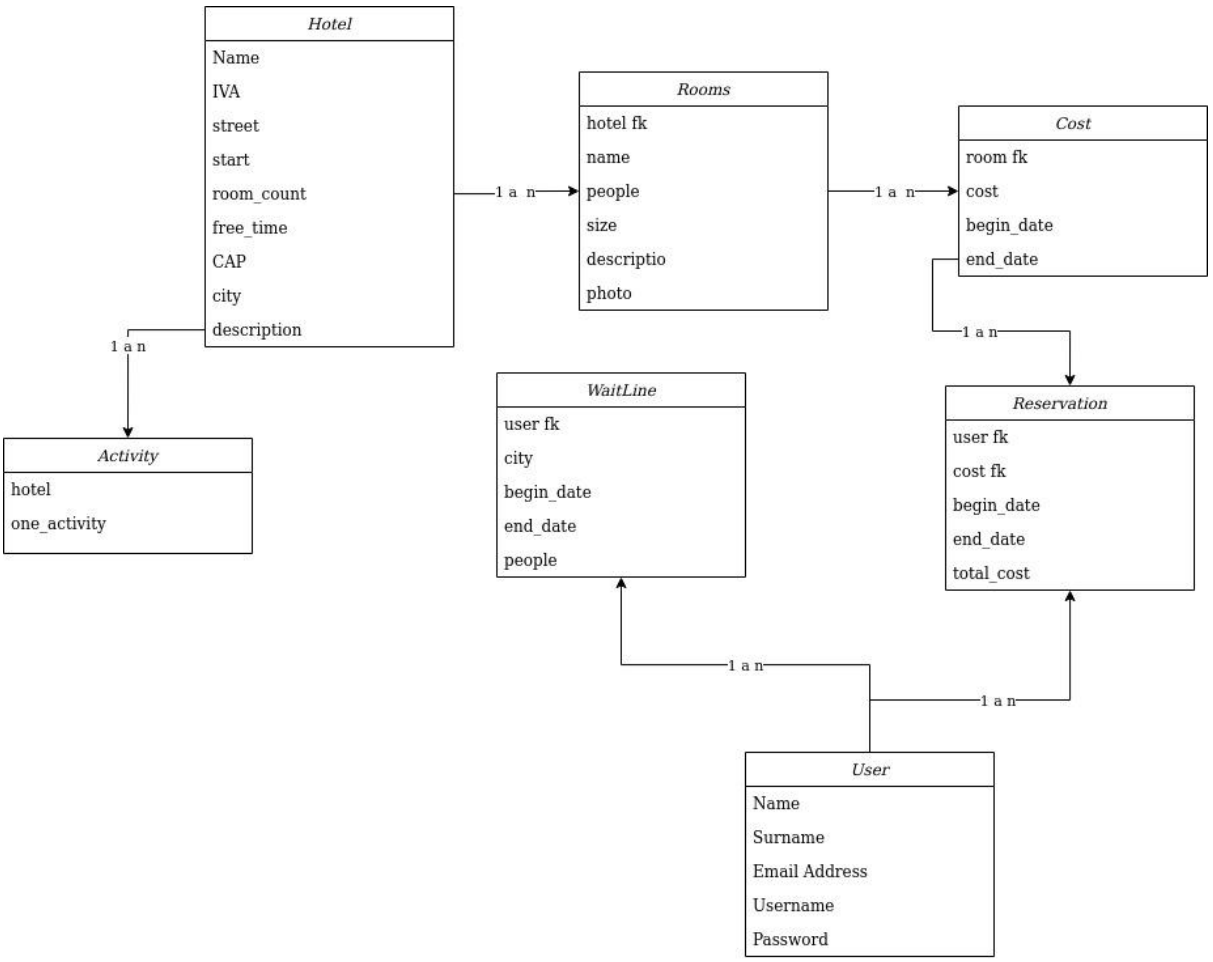
È un modello con cui l'utente non interagisce direttamente ma che, a creazione di hotel, filtra il campo di testo free_time e salva ogni attività con hotel come foreign key. In questo modo è possibile aggiornare le attività disponibili in maniera dinamica, correndo però il rischio di user error.

## Reservation

Ogni utente può avere n prenotazioni, e ogni prenotazione salva il periodo di permanenza e il costo da cui deriva. In questo modo diventa molto facile fare un query di ricerca tra Cost e Reservation servendosi degli id di cost.

## WaitLine

Permette di salvare un periodo che non ha nessuna camera disponibile. Periodicamente una task fa una query di ricerca su questa tabella e nel caso in cui ritorni un risultato manda una mail all'utente e cancella tale utente dal database.




# Tecnologie utilizzate

Django full stack, con django template language e css. Boostrap e ion-icon hanno permesso uno sviluppo veloce e con un design solido.

# Librerie

Per poter realizzare il sistema di attesa è stata impiegata la libreria Celery.

## Celery

Celery si può installare tramite pip install celery. Nel caso in cui celery crei problemi di compatibilità installare setuptools==58, tramite pip.
Le configurazioni a livello di progetto django vanno definite in un loro file a parte celery.py.

Una volta definita la configurazione base è possibile creare file tasks.py all'interno dell'app necessaria, per poi decorarli con  @share_task e al loro interno definire il set di operazioni da fare.

Dentro settings.py definire le variabili globali di celery e creare la costante CELERY_BEAT_SCHEDULE che periodicamente avvia una task.

Adesso abbiamo una task che fa una query su una tabella e manda una mail se si libera un posto e una configurazione che periodicamente lancia tale task.
Infine installare e avviare il servizio rabbitmq e lanciare i seguenti comandi in 2 terminali diversi:

- celery -A DumitruFrunza135801 worker -l INFO
- celery -A DumitruFrunza135801 beat -l INFO

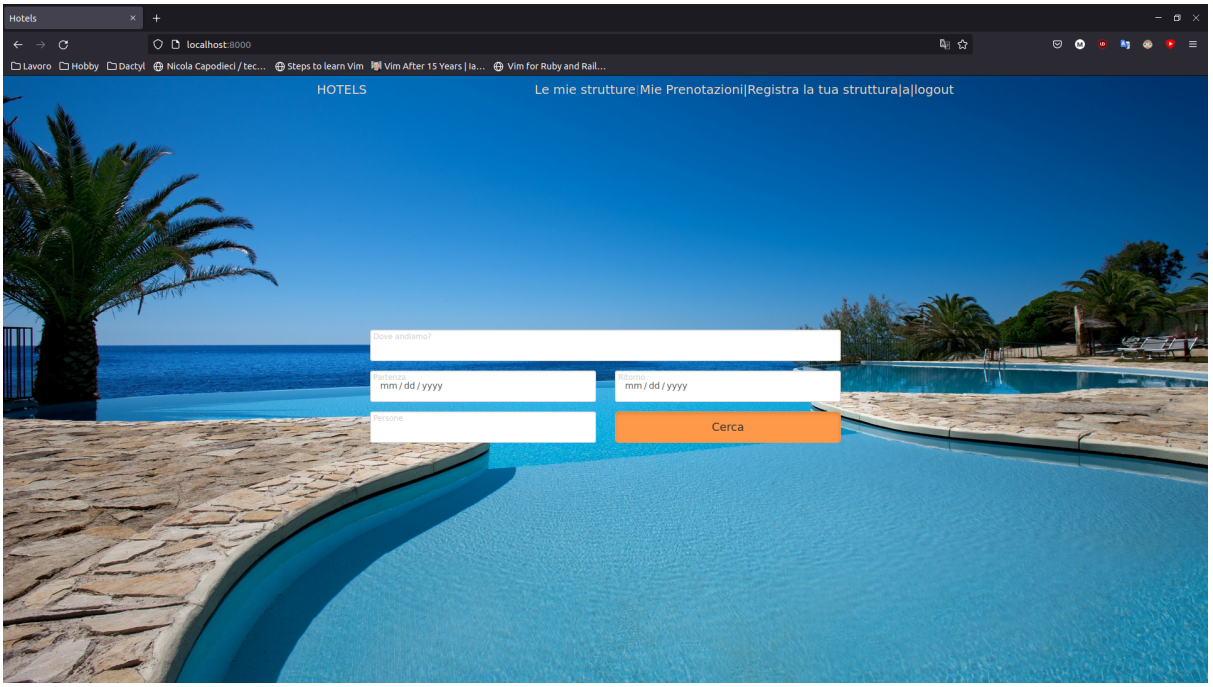
# App

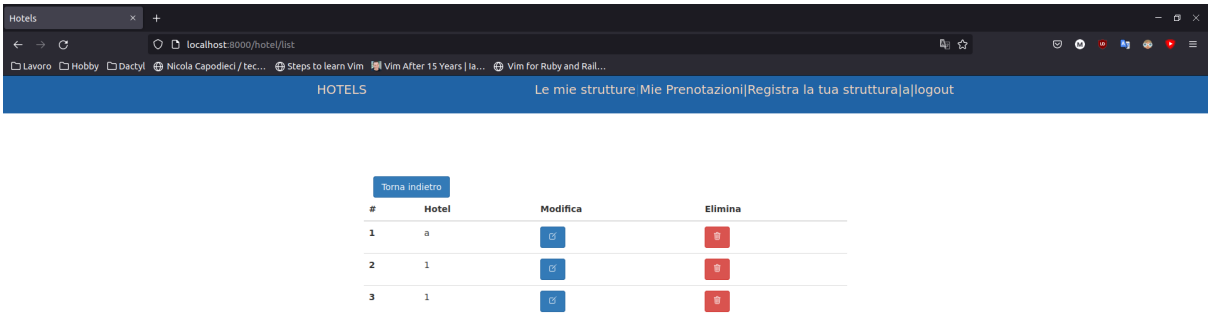
Gran parte del progetto è contenuto dentro l'app hotel, ho deciso di dividere solo il login e il home (che include anche la query di ricerca)

# Test

Sono state testate alcune view e l'effettiva creazione di un oggetto di tipo database.

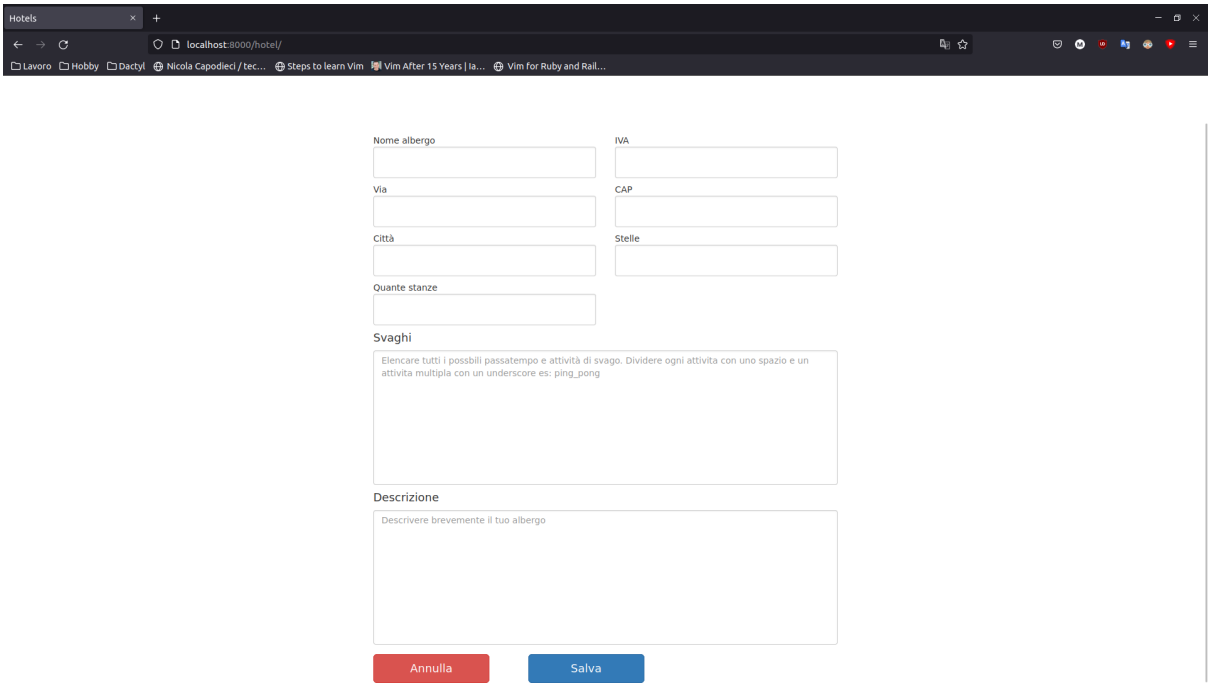
# Viste finali

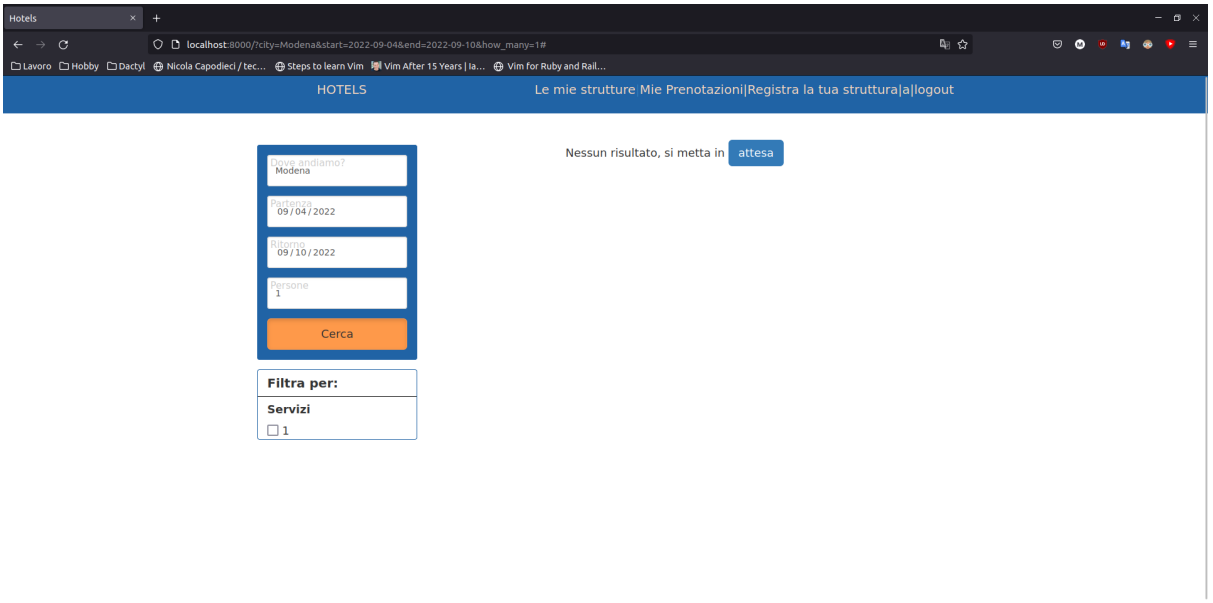


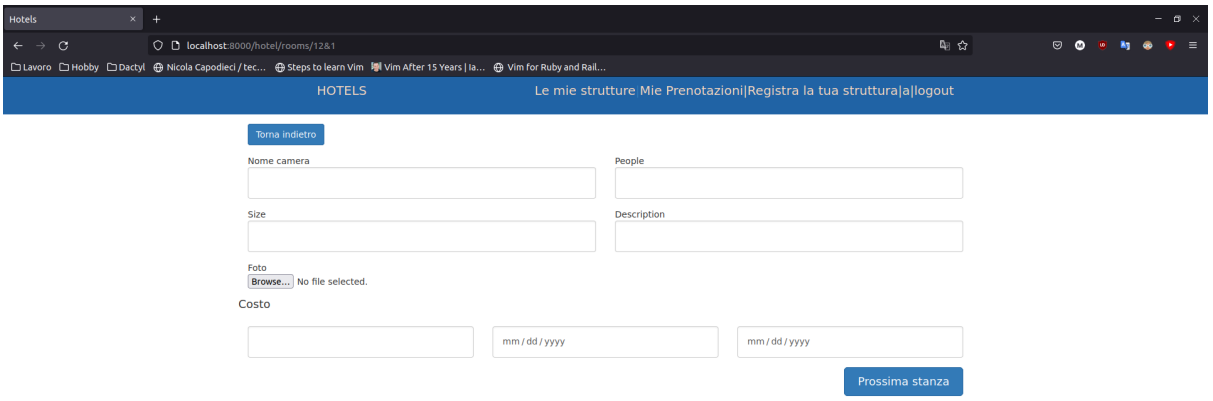




Torna indietro

| # | Hotel | Modifica | Elimina |
|---|---|---|---|
| 1 | a | | |
| 2 | 1 | | |
| 3 | 1 | | |


Hotels
localhost:8000/hotel/
Lavoro
Hobby
Dactyl
Nicola Capodieci / tec...
Steps to learn Vim
Vim After 15 Years | la...
Vim for Ruby and Rail...
Nome albergo
IVA
Via
CAP
Città
Stelle
Quante stanze
Svaghi
Elencare tutti i possbili passatempo e attività di svago. Dividere ogni attivita con uno spazio e un attivita multipla con un underscore es: ping_pong
Descrizione
Descrivere brevemente il tuo albergo
Annulla
Salva




# Difficoltà

Data la mia scarsa esperienza con javascript non sono riuscito a rendere dinamico quanto avevo pianificato l'inserimento di contenuti.
Celery è un libreria con una documentazione meno chiara e esauriente quanto quella di django e richiede una notevole configurazione, ha portato via molto tempo per una feature che mi aspetterei già integrata in django.