

# SDD

## (System Design Document)

### INDICE

1. [Obiettivi Generali](#)
2. [Architettura Software Corrente](#)
3. [Obiettivi di Progettazione](#)
4. [Architettura Software Proposta](#)
  - 4.1. [Panoramica](#)
  - 4.2. [Decomposizione in sottosistemi](#)
    - 4.2.1. [Suddivisione degli oggetti all'interno dei sottosistemi](#)
  - 4.3. [Mappatura Hardware/Software](#)
  - 4.4. [Gestione dei Dati Persistenti – Progetto del Database](#)
    - 4.4.1. [Modello ER](#)
    - 4.4.2. [Descrizione delle Tabelle](#)
  - 4.5. [Accesso al Sistema](#)

Documento redatto dal Team Mars

Maragliano Francesco

Alabiso Davide

Romano Sonny

Sanguedolce Giuseppe

# 1 Obiettivi Generali

Lo scopo del sistema proposto è quello di informatizzare le principali attività, svolte dal proprietario di una “Casa Vacanze” o B&B, atte a gestire tutte le operazioni di promozione, prenotazione, pagamento e adempimento a doveri burocratici dei propri annunci. Inoltre, il sistema permette la comunicazione fra cliente, intenzionato ad affittare, e proprietario del relativo annuncio in modo rapido e semplice.

## 2 Architettura Software Corrente

Si presuppone che non esista alcun software con le stesse funzionalità attualmente adottato dal committente del lavoro, e che quindi le attività che il Sistema ha l’obbiettivo di informatizzare fossero svolte manualmente.

## 3 Obiettivi di Progettazione

- Il Sistema deve impedire l’inserimento di input non corretti da parte degli utenti.
- Il Sistema deve essere accessibile da qualunque dispositivo grazie ad un sistema che permetta l’hosting su Server senza quindi necessitare file in locale.
- Il Sistema deve impedire a coloro i quali usufruiscono di quest’ultimo l’accesso a sezioni di cui non hanno i permessi.

## 4 Architettura Software Proposta

### 4.1 Panoramica

Per il Sistema è stata scelta un’architettura Three-Tier.

Lo “Storage Layer” si occupa dell’immagazzinamento dei dati, del loro inserimento e recupero ed è rappresentato dal lato API – Express e dal Database del Sistema.

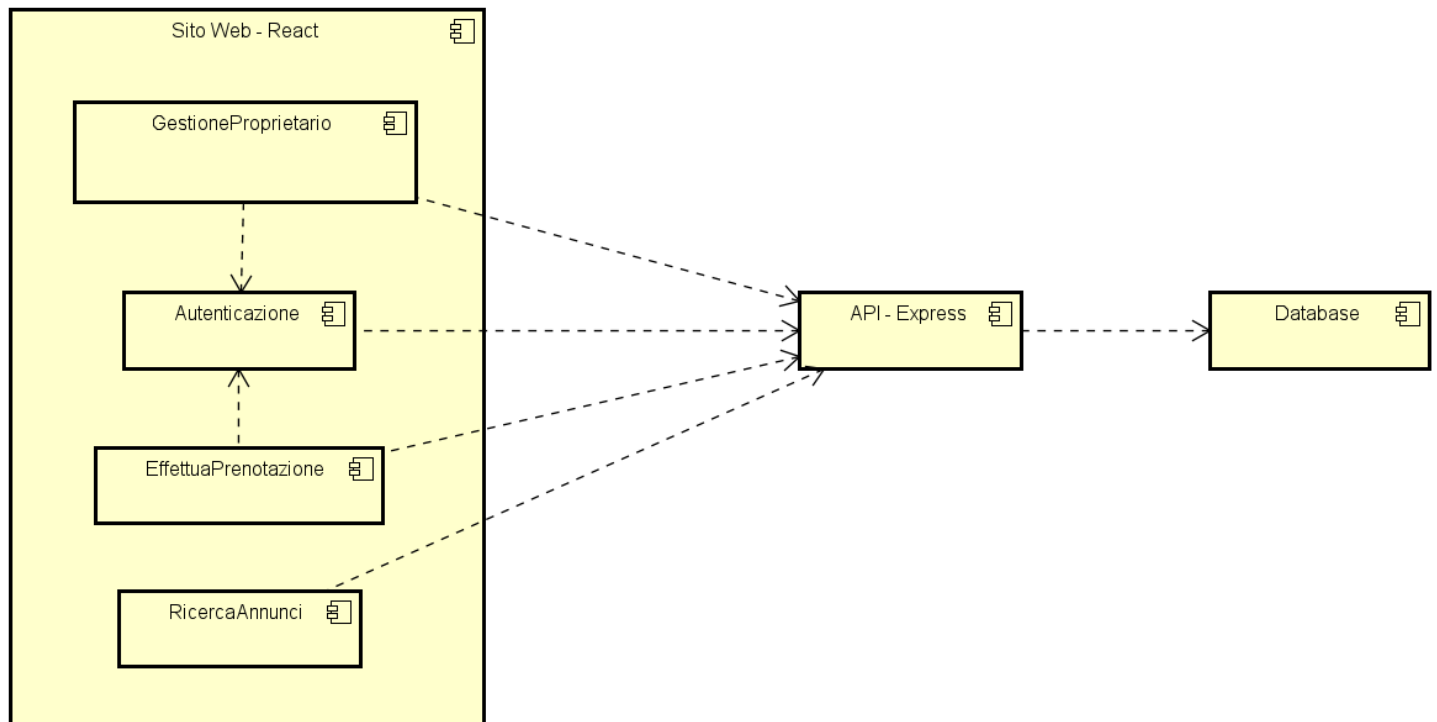
“L’ Application Logic Layer” si occupa elaborare i processi e compiere la maggior parte delle operazioni, inoltre collega “L’ Interface Layer” con lo “Storage Layer”. É rappresentato dai componenti React.

“L’ Interface Layer” si occupa dell’interazione con l’utente attraverso Bottoni, Form e Finestre. É rappresentato dalle sezioni HTML nel “renderizzati” dai componenti React.

## 4.2 Decomposizione in Sottosistemi

Attraverso l'utilizzo dei 'component diagram' di UML, viene rappresentata la decomposizione del Sistema in sottogruppi logicamente accomunati.

Ciò rende più flessibile il Sistema e marca maggiormente la struttura dell'architettura Three-Tier scelta.



Il sottosistema “Sito Web – React” racchiude tutte le componenti grafiche e di interazione dell’utente col Sistema. È, perciò, stato diviso ulteriormente nei sottosistemi

“GestioneProprietario”, “Autenticazione”, “EffettuaPrenotazione” e “RicercaAnnunci”.

“GestioneProprietario” permette ad un utente con privilegi da ‘host’ di gestire tutte le operazioni relative alla gestione di un annuncio.

“Autenticazione” permette ad un visitatore di effettuare le operazioni per accedere al sito come utente.

“EffettuaPrenotazione” racchiude l’operazione per permettere ad un utente di affittare un appartamento.

“RicercaAnnunci” consente ad un visitatore di cercare e visualizzare tra gli annunci disponibili.

Il sottosistema “API - Express” contiene le operazioni per interagire col “Database”.

Infine, il “Database” contiene le tabelle atte a contenere tutti i dati utili. Per un maggiore approfondimento circa la sua struttura, si rimanda al paragrafo 4.4.

“GestioneProprietario” è stato ulteriormente suddiviso per garantire una migliore visione delle operazioni disponibili.



Per dipendenza dal padre, anche questi sottosistemi richiedono il collegamento ad “Autenticazione” e “API - Express”.

“GestioneAnnunci” permette al proprietario di inserire, modificare ed eliminare annunci, oltre a permettergli di visualizzare una lista di quelli già inseriti.

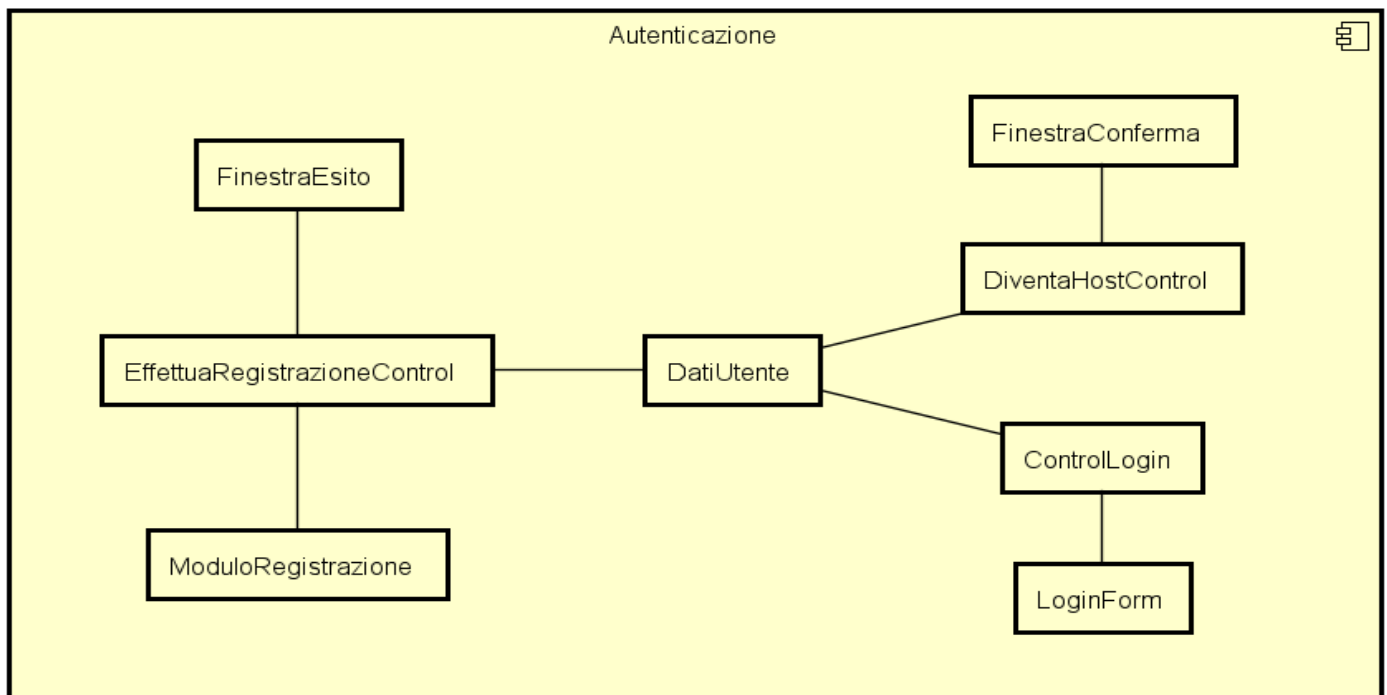
“GestionePrenotazioni” permette al proprietario di visualizzare le attuali prenotazioni pendenti e di confermarle o rifiutarle. Inoltre, gli permette di visualizzare i guadagni delle prenotazioni relative ai suoi annunci.

“Questura” permette al proprietario di comunicare alla Questura la presenza degli ospiti relativi alle prenotazioni effettuate attraverso il Sistema.

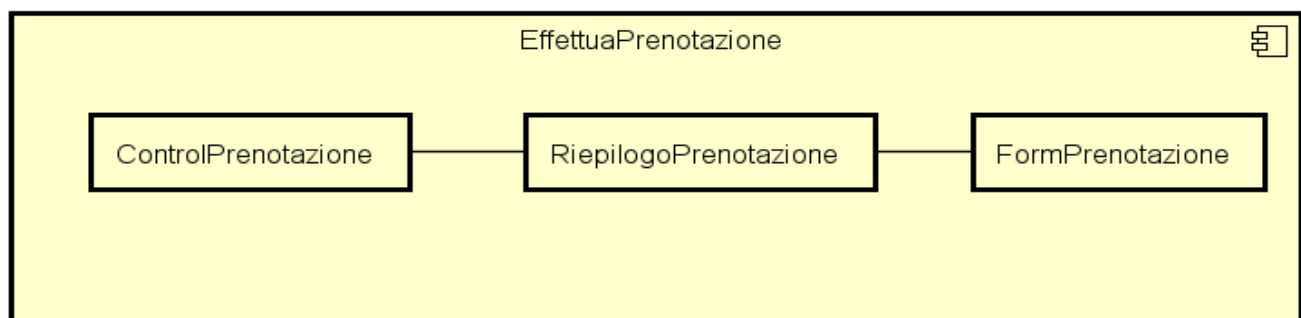
“TasseSoggiorno” permette al proprietario di pagare le tasse di soggiorno al Comune relative alle prenotazioni effettuate attraverso il Sistema.

## 4.2.1 Suddivisione degli oggetti all'interno dei sottosistemi

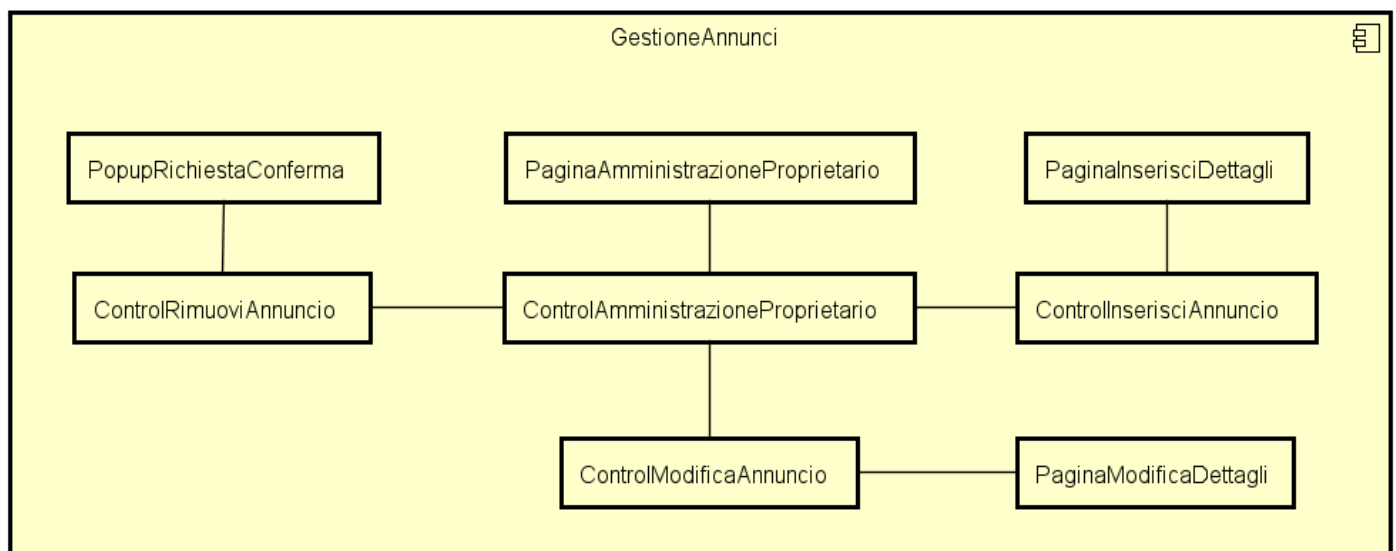
### Autenticazione



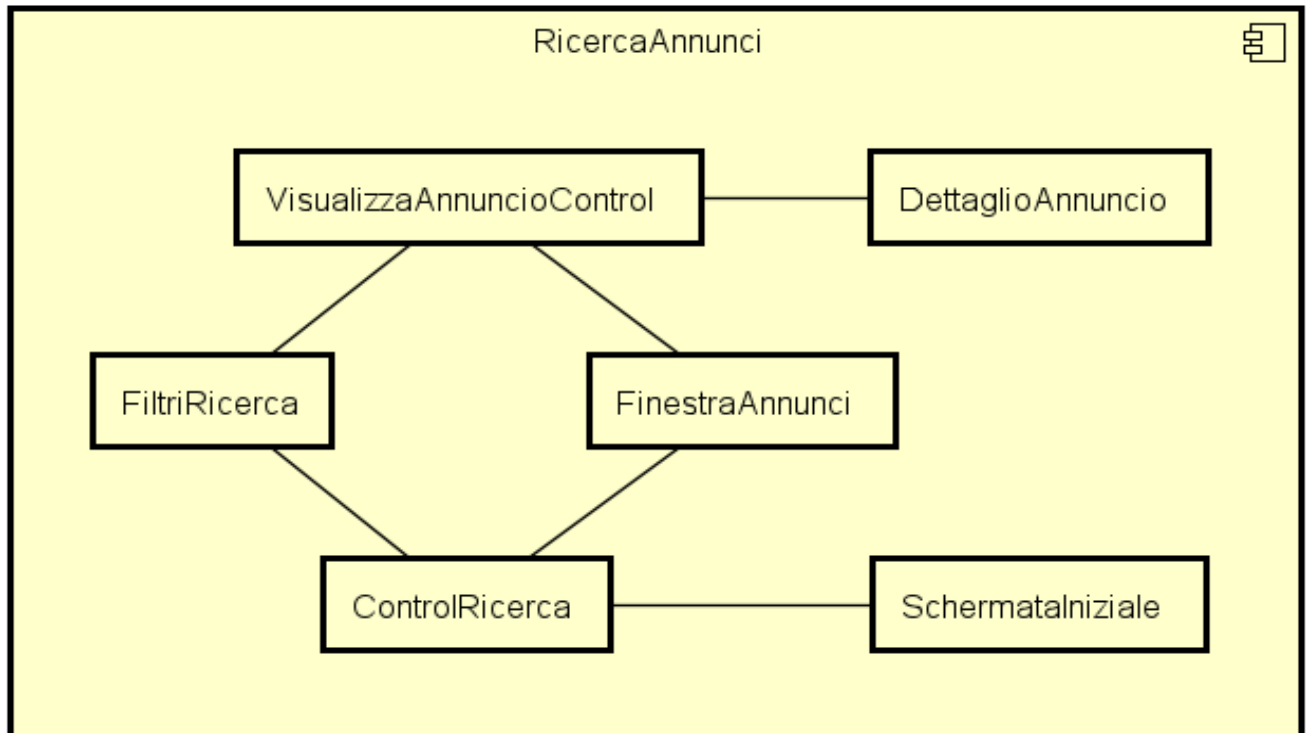
### Effettua Prenotazione



### Gestione Annunci



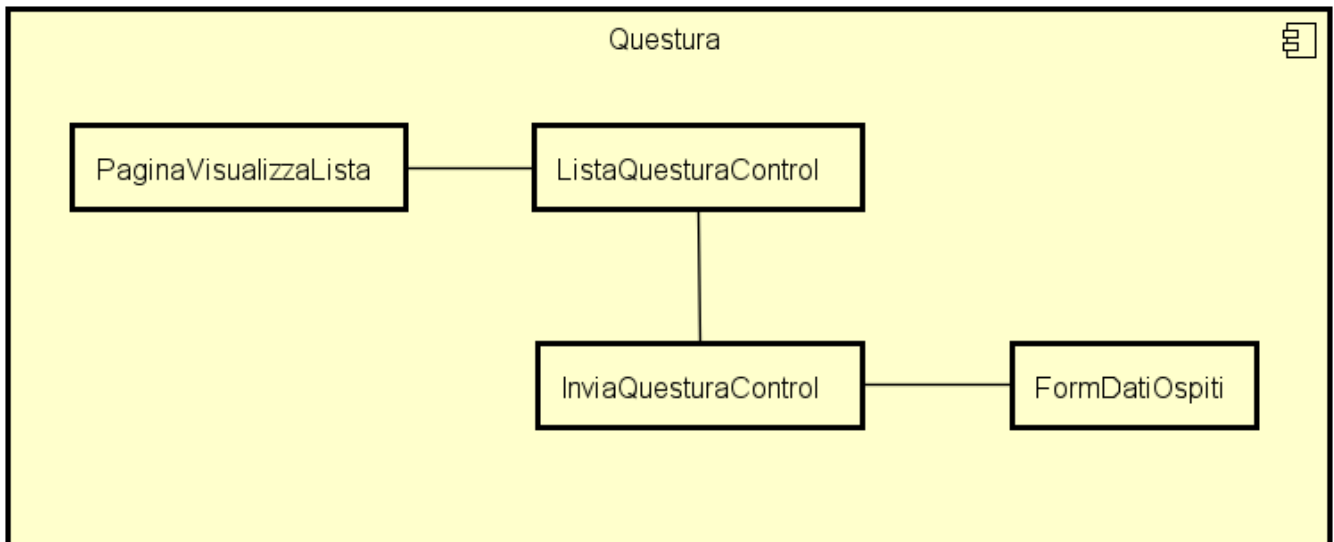
## Ricerca Annunci



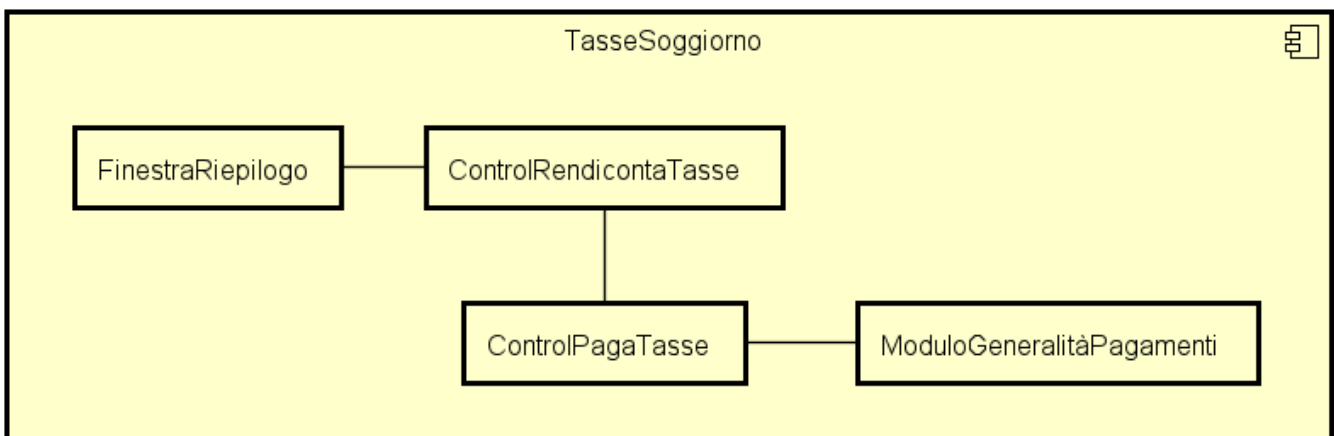
## Gestione Prenotazioni



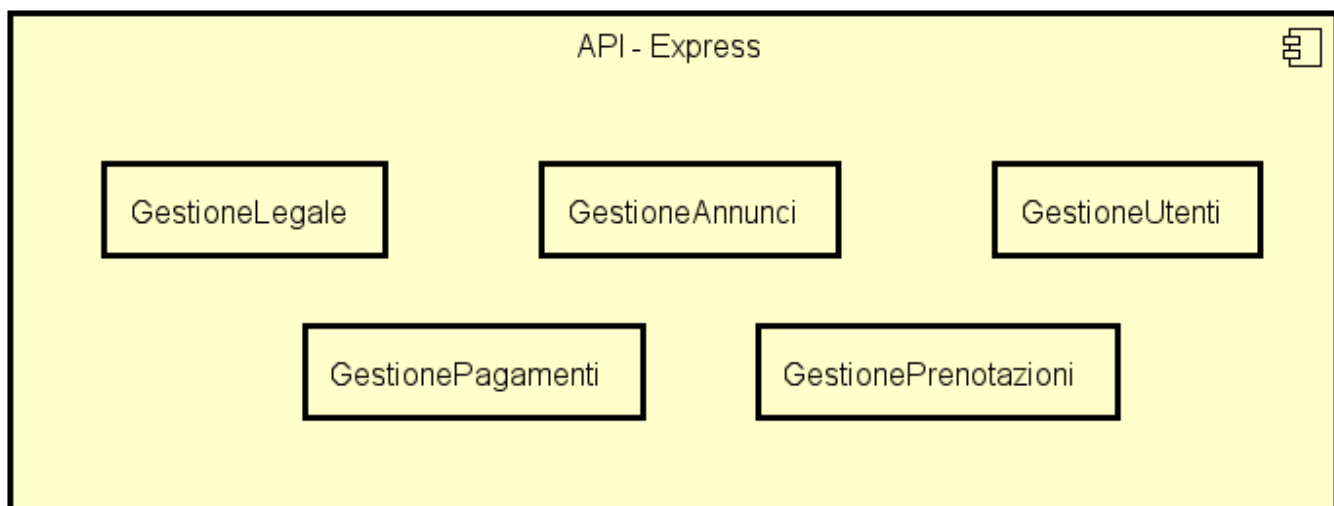
## Questura



## TasseSoggiorno



## API - Express



## 4.3 Mappatura Hardware/Software

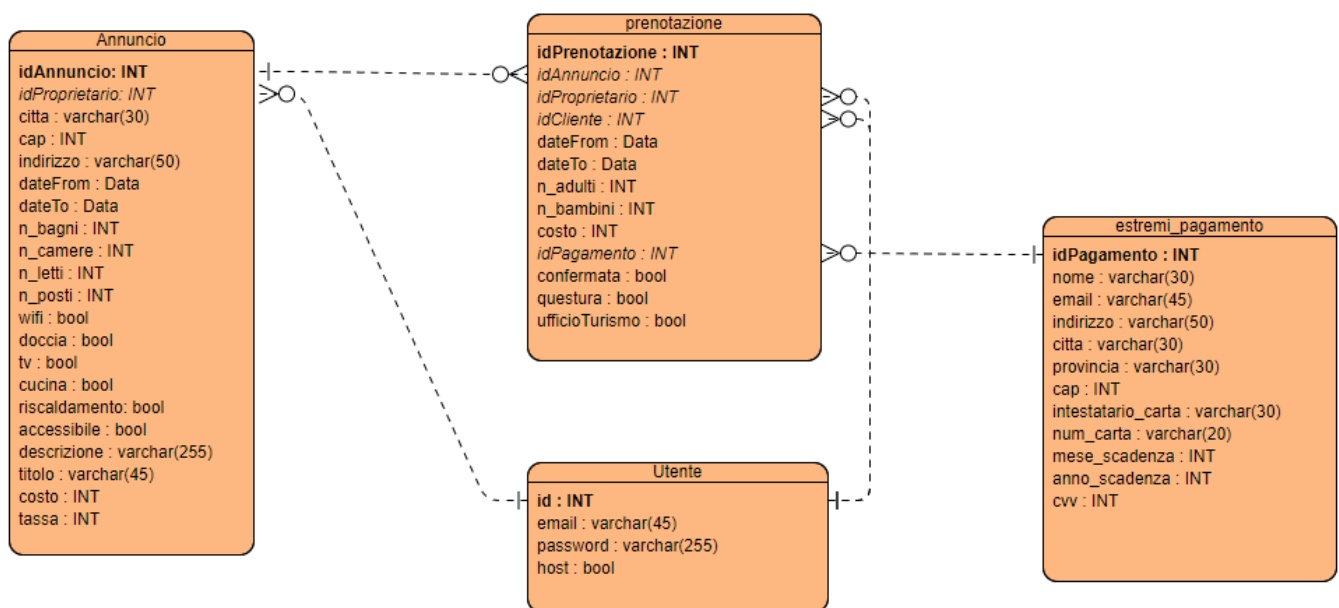
Non sono necessari particolari requisiti Hardware per poter usufruire del Sistema.

Grazie al servizio di “hosting” offerto da Heroku™ è possibile utilizzare il Sistema da qualunque dispositivo abbia accesso ad Internet, sia esso Desktop o Mobile.

In caso si voglia utilizzare la versione locale del Sistema, saranno necessari almeno 150Mb di spazio libero su disco e NodeJS installato nel dispositivo. Ciò si riferisce solo alla versione Desktop.

## 4.4 Gestione dei dati persistenti – Progetto del Database

### 4.4.1 Modello ER



### Vincoli di Tupla:

- La chiave primaria idAnnuncio della tabella Annuncio non può assumere mai valore null.
- La chiave primaria idPrenotazione della tabella prenotazione non può assumere mai valore null.
- La chiave primaria id della tabella Utente non può assumere mai valore null.
- La chiave primaria idPagamento della tabella estremi\_pagamento non può assumere mai valore null.
- Il campo password della tabella Utente deve essere composto da almeno 8 caratteri, di cui un carattere maiuscolo, uno minuscolo e un numero.
- Il campo dateFrom deve essere sempre minore in ordine temporale del campo dateTo nella tabella Annunci.
- Il campo dateTo non si deve mai distanziare temporalmente per un totale di giorni superiore a 28, rispetto al campo dateFrom, nella tabella Annunci.
- Il campo titolo della tabella Annunci deve essere composto al più da 45 caratteri.
- Il campo cap della tabella Annunci deve essere composto da 5 caratteri.
- I campi idProprietario e idCliente della tabella prenotazioni, non possono fare riferimento ad una stessa istanza della tabella Utente.
- Il campo num\_carta della tabella estremi\_pagamento deve essere composto da 16 caratteri.
- Il campo cvv della tabella estremi\_pagamento deve essere composto da 3 caratteri.
- Il campo cap della tabella estremi\_pagamento deve essere composto da 5 caratteri.



- Una istanza della tabella Prenotazioni avente riferimento esterno idPagamento non nullo, deve avere necessariamente il campo confermata settato a true.
- Per una istanza della tabella prenotazione avente riferimento esterno tramite il campo idAnnuncio alla tabella Annunci, la somma dei campi n\_adulti + n\_bambini nella tabella prenotazione, deve essere minore o uguale al campo n\_posti della tabella annunci.
- Nella tabella prenotazione, affinché i campi questura o ufficioTurismo o entrambi possano essere settati a true, si presuppone che il campo confermata sia settato a true.

## 4.4.1 Descrizione delle tabelle

### Annuncio

Nome colonna	Tipo	Vincoli	Descrizione
idAnnuncio	int	PK, AI, NOT NULL	Chiave primaria che identifica univocamente tutti gli annunci
idProprietario	int	NOT NULL	Riferimento ad una tupla della tabella Utenti. Identifica il proprietario dell'annuncio
citta	varchar(30)	NOT NULL	Identifica il nome della città dove si trova l'appartamento
cap	int	NOT NULL	Identifica il cap della città dove si trova l'appartamento
Indirizzo	varchar(50)	NOT NULL	Identifica l'indirizzo della città dove si trova l'appartamento
dateFrom	data	NOT NULL	Identifica la data di inizio disponibilità dell'appartamento ad essere affittato
dateTo	data	NOT NULL	Identifica la data di fine disponibilità dell'appartamento ad essere affittato
n_bagni	int	NOT NULL	Identifica il numero di bagni presenti nell'appartamento
n_camere	int	NOT NULL	Identifica il numero di camere presenti nell'appartamento
n_letti	int	NOT NULL	Identifica il numero di posti letto presenti nell'appartamento
n_posti	int	NOT NULL	Identifica il numero massimo di ospiti che può ospitare l'appartamento
wifi	bool	NOT NULL	Identifica la presenza o meno del wifi nell'appartamento
doccia	bool	NOT NULL	Identifica la presenza o meno della doccia nell'appartamento
tv	bool	NOT NULL	Identifica la presenza o meno del wifi nell'appartamento
cucina	bool	NOT NULL	Identifica la presenza o meno della cucina nell'appartamento
riscaldamento	bool	NOT NULL	Identifica la presenza o meno del riscaldamento nell'appartamento
accessibile	bool	NOT NULL	Identifica se l'appartamento è accessibile a persone disabili o meno
descrizione	varchar(255)	NOT NULL	Elemento testuale contenente la descrizione dell'annuncio e/o dettagli aggiuntivi
titolo	varchar(45)	NOT NULL	Elemento testuale contenente il titolo dell'annuncio

costo	int	NOT NULL	Identifica il costo giornaliero da pagare per poter affittare l'appartamento
tassa	int	NOT NULL	Identifica l'ammontare della tassa da versare all'ufficio turismo per l'appartamento

## Prenotazione

Nome colonna	Tipo	Vincoli	Descrizione
idPrenotazione	int	PK, AI, NOT NULL	Chiave primaria che identifica univocamente tutte le prenotazioni
idAnnuncio	int	NOT NULL	Riferimento ad una tupla della tabella Annuncio. Identifica l'annuncio cui il cliente ha richiesto la prenotazione
idProprietario	int	NOT NULL	Riferimento ad una tupla della tabella Utenti. Identifica il proprietario dell'annuncio prenotato
idCliente	int	NOT NULL	Riferimento ad una tupla della tabella Utenti. Identifica il cliente che ha prenotato l'annuncio
dateFrom	data	NOT NULL	Identifica la data di inizio della prenotazione dell'appartamento da parte del cliente
dateTo	data	NOT NULL	Identifica la data di fine della prenotazione dell'appartamento da parte del cliente
n_adulti	int	NOT NULL	Identifica il numero di ospiti adulti indicati dal cliente all'atto della prenotazione
n_bambini	int	NOT NULL	Identifica il numero di ospiti bambini indicati dal cliente all'atto della prenotazione
costo	int	NOT NULL	Identifica il prezzo totale pagato dal cliente all'atto della prenotazione dell'appartamento
idPagamento	int	UNIQUE, NOT NULL	Riferimento ad una tupla della tabella estremi_pagamento. Identifica i dati degli estremi di pagamento specificati dal cliente all'atto della prenotazione
confermata	bool	NOT NULL	Identifica se la prenotazione richiesta dal cliente è stata accettata dal proprietario
questura	bool	NOT NULL	Identifica, per una prenotazione confermata, se il proprietario vi ha comunicato o meno la presenza degli ospiti alla questura
ufficioTurismo	bool	NOT NULL	Identifica, per una prenotazione confermata, se il proprietario vi ha versato o meno le tasse all'ufficio del turismo

## Utente

Nome colonna	Tipo	Vincoli	Descrizione
id	int	PK, AI, NOT NULL	Chiave primaria che identifica univocamente tutti gli account degli utenti della piattaforma
email	Varchar(45)	NOT NULL	Identifica l'indirizzo e-mail associato all'account dell'utente
password	Varchar(255)	NOT NULL	Identifica la password associata all'account dell'utente. Per questioni di sicurezza la password viene memorizzata in forma cifrata
host	bool	NOT NULL, DEFAULT 0	Identifica o meno se l'account dell'utente ha ottenuto i privilegi da host

## estremi\_pagamento

Nome colonna	Tipo	Vincoli	Descrizione
idPagamento	int	PK, AI, NOT NULL	Chiave primaria che identifica univocamente tutti gli estremi di pagamento specificati dagli utenti all'atto delle prenotazioni
nome	varchar(30)	NOT NULL	Identifica il nome e cognome di fatturazione specificato dal cliente all'atto della prenotazione
email	varchar(45)	NOT NULL	Identifica l'e-mail di fatturazione specificata dal cliente all'atto della prenotazione
indirizzo	varchar(50)	NOT NULL	Identifica l'indirizzo di fatturazione specificato dal cliente all'atto della prenotazione
citta	varchar(30)	NOT NULL	Identifica la città di fatturazione specificata dal cliente all'atto della prenotazione
provincia	varchar(30)	NOT NULL	Identifica la provincia di fatturazione specificata dal cliente all'atto della prenotazione
cap	int	NOT NULL	Identifica il cap di fatturazione specificato dal cliente all'atto della prenotazione
intestatario_carta	varchar(30)	NOT NULL	Identifica il nome e cognome dell'intestatario della carta con cui il cliente intende pagare la prenotazione
num_carta	varchar(20)	NOT NULL	Identifica il numero della carta con cui il cliente intende pagare la prenotazione
mese_scadenza	int	NOT NULL	Identifica il mese di scadenza della carta con cui il cliente intende pagare la prenotazione
anno_scadenza	int	NOT NULL	Identifica l'anno di scadenza della carta con cui il cliente intende pagare la prenotazione
cvv	int	NOT NULL	Identifica il cvv della carta con cui il cliente intende pagare la prenotazione

## 4.5 Accesso al Sistema

L'accesso al Sistema è completamente libero, previa registrazione e successiva autenticazione.

L'accesso come visitatore è altamente limitato e consente la semplice ricerca e visualizzazione degli annunci.

La registrazione come utente permette, invece, anche di poter prenotare.

Attraverso la procedura “Diventa Host” è poi possibile ottenere un accesso completo a tutte le funzionalità del Sistema. Per eseguire la suddetta procedura non è necessario possedere nessun requisito particolare se non il possesso di un account.