



UNIVERSITÀ DEGLI STUDI DI PALERMO

System Design Document

Progetto a cura di:

Fabio Villa
Riccardo Licari
Giovanni Bernardo
Emanuele Adolfo Ferrara

Indice

1. Obiettivi generali
2. Sistema Corrente
3. Architettura software proposta
 - 3.1. Decomposizione in sottosistemi
 - 3.1.1. Suddivisione degli oggetti all'interno dei sottosistemi e delle componenti
 - 3.2. Gestione dei dati persistenti - Progetto del Database
 - 3.2.1. Modello E-R
 - 3.2.2. Modello Relazionale
 - 3.2.3. Considerazioni sul database
 - 3.2.4. Descrizione delle tabelle
 - 3.3. Accesso al sistema e sicurezza
 - 3.3.1. Controlli di Integrità e Sicurezza del Sistema

INTRODUZIONE

1. Obiettivi generali

Il software si occuperà di gestire una piattaforma di prenotazione per strutture turistiche tipo casa vacanza o B&B, permettendo agli utenti di prenotare un alloggio in una delle strutture che si appoggiano alla piattaforma. Il software prevede una distinzione tra i diversi tipi di utilizzatori della piattaforma: Utente, Utente registrato, Amministratore. Ogni fruitore della piattaforma avrà a disposizione diverse funzionalità, in base al suo ruolo.

2. Sistema corrente

Si suppone che, attualmente, l'azienda non abbia adottato alcun tipo di software con le stesse funzionalità di questo proposto.

3. Architettura software proposta

Per il sistema è stata scelta un'architettura **Three-Tier** ibrida con un'architettura **Client-Server**

Architettura Three-Tier:

- 1) **Interface Layer**: è il livello che comprende le pagine web generate dinamicamente; queste dialogano e scambiano informazioni con il server attraverso le API fornite.
- 2) **Application Logic Layer**: è il livello che fornisce le funzionalità base del sistema.
- 3) **Storage Layer**: è il livello che mantiene e gestisce la conoscenza dei dati del sistema attraverso l'utilizzo di un DBMS.

L'accesso al database avviene utilizzando il pattern architetturale **Data Access Object**, il quale fornisce le funzionalità di accesso ai dati.

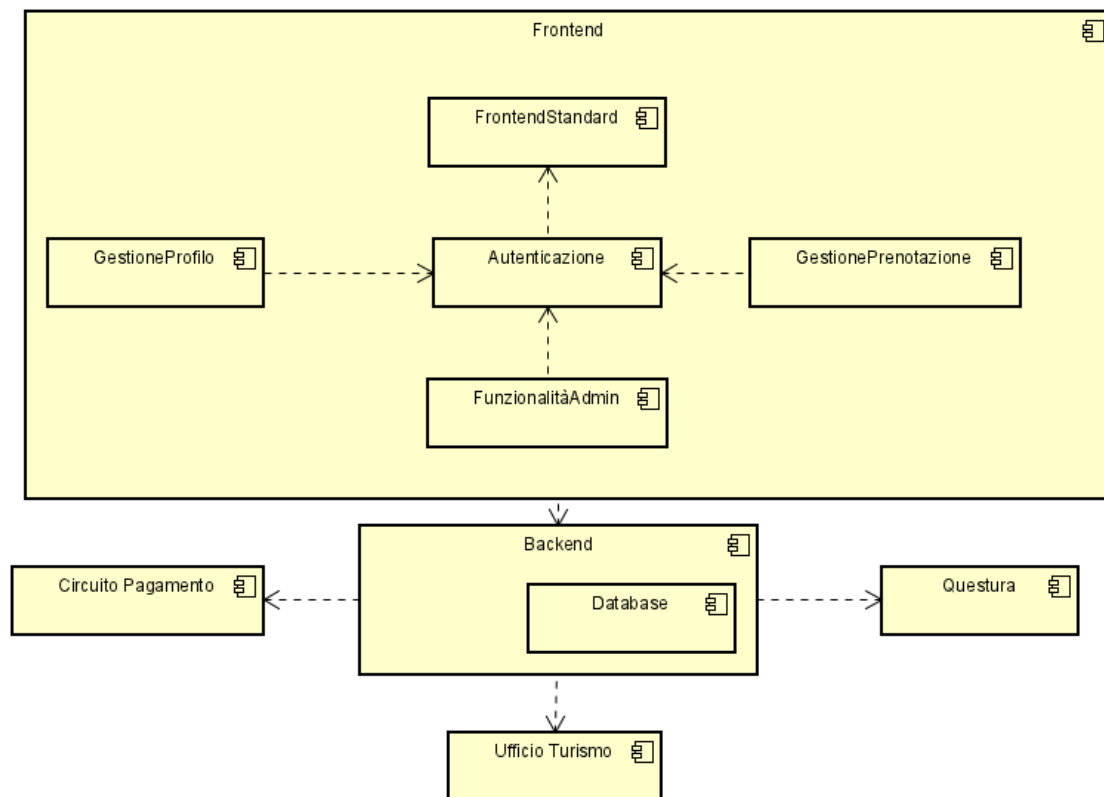
Architettura Client-Server

L'architettura del sistema prevede l'utilizzo di **due server**:

- 1) **Frontend** : creato attraverso l'utilizzo di ReactJS. Fornisce le pagine web e inoltra le richieste HTTPS al server di Backend
- 2) **Backend**: creato attraverso l'utilizzo di Express. Riceve le richieste HTTPS dal Frontend e fornisce le informazioni/funzionalità richieste.
Comunica con il DBMS, con l'Ufficio Turismo, con l'ufficio della questura. Il server fornisce ai client (Web browsers) la possibilità di interfacciamento attraverso delle pagine web.

I **client** rappresentano i terminali di accesso al sistema per i diversi utilizzatori del software. Per "terminali di accesso" si intende un qualsiasi dispositivo in grado di supportare e utilizzare un web browser (Pc, Tablet, Smartphone ecc..)

3.1 Decomposizione in sottosistemi

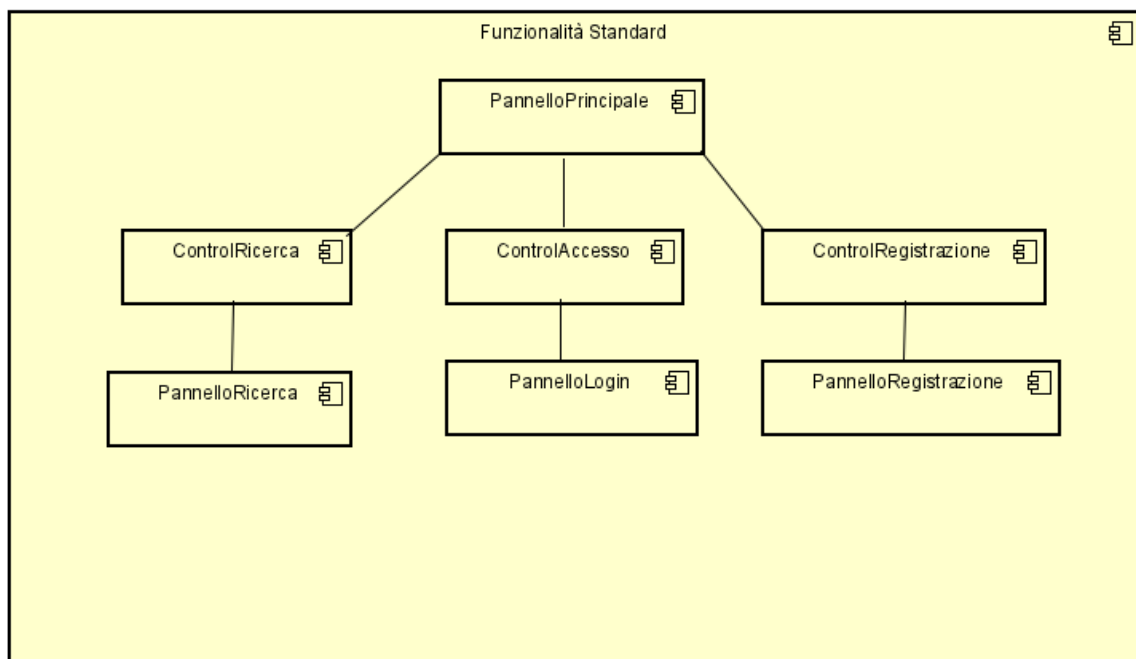


Il Front-end è suddiviso in Funzionalità Standard, quelle accessibili da tutti come la ricerca, il crearsi un account o poter effettuare un login, infatti chi effettua un login può essere identificato come amministratore o come utente normale che usufruisce il sito, l'utente ha diritto ad azioni come l'effettuare una prenotazione o gestire il suo profilo, oppure ancora potendo vedere e gestire le prenotazioni, sia che esse siano la registrazione di un alloggio, vedere le prenotazioni che quest'ultimo ha ricevuto ma anche vedere e poter gestire le prenotazioni che ha effettuato.

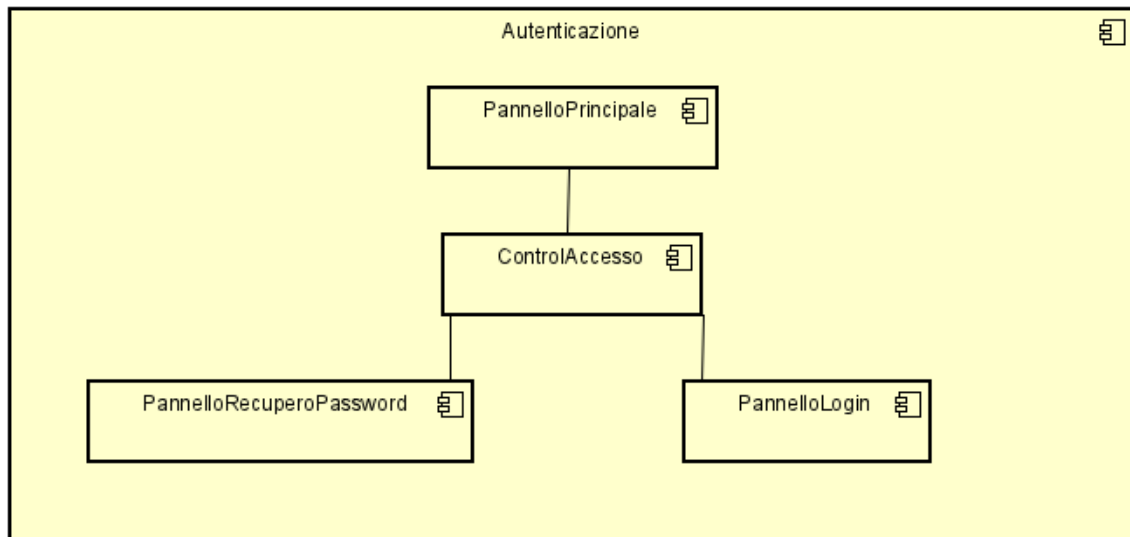
Chi invece viene riconosciuto come amministratore ha accesso a quella serie di funzionalità Admin, ovvero poter eliminare altri utenti e poter eliminare alloggi di specifici utenti

3.1.1 Suddivisione degli oggetti all'interno dei sottosistemi e delle componenti

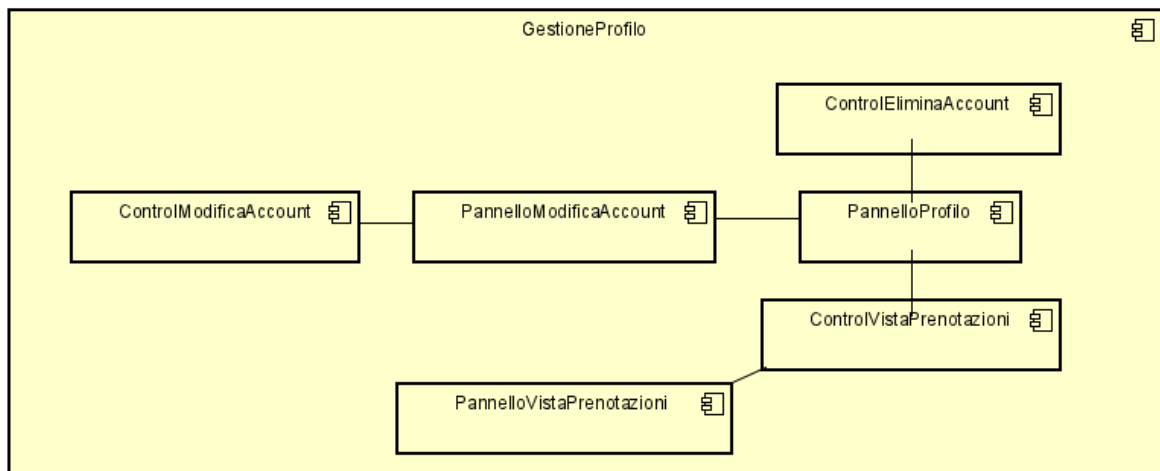
Funzionalità standard



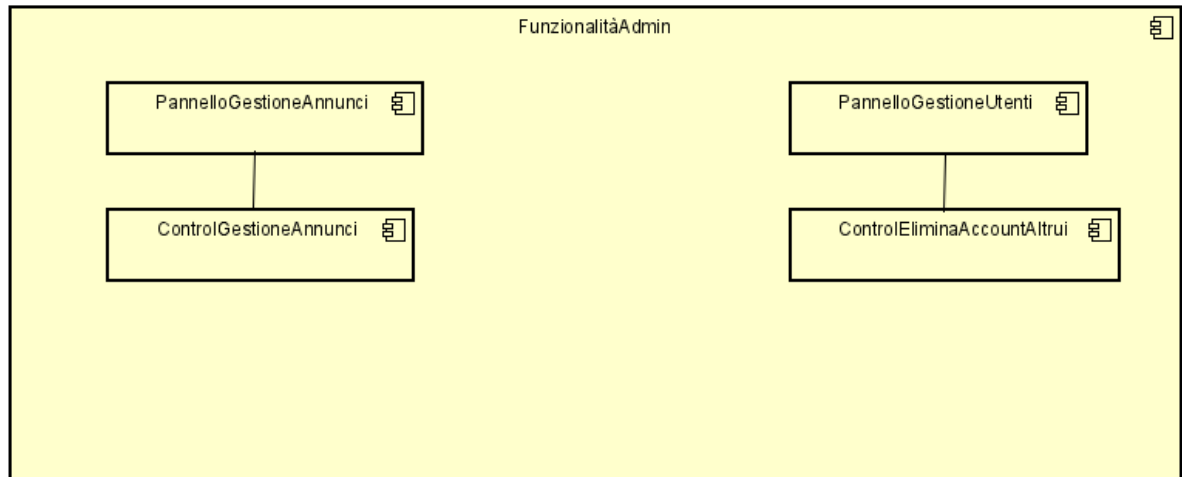
Autenticazione



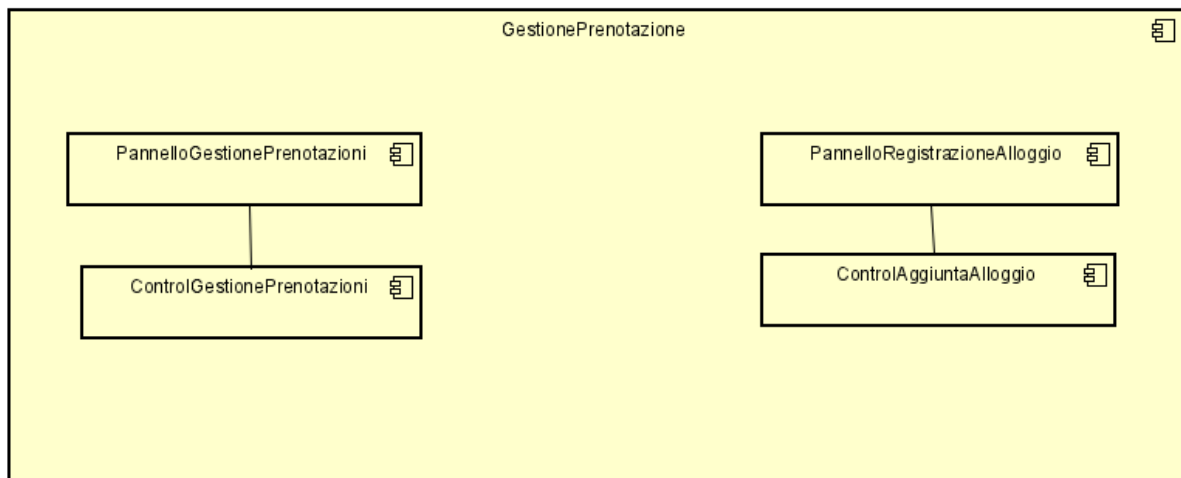
Gestione Profilo

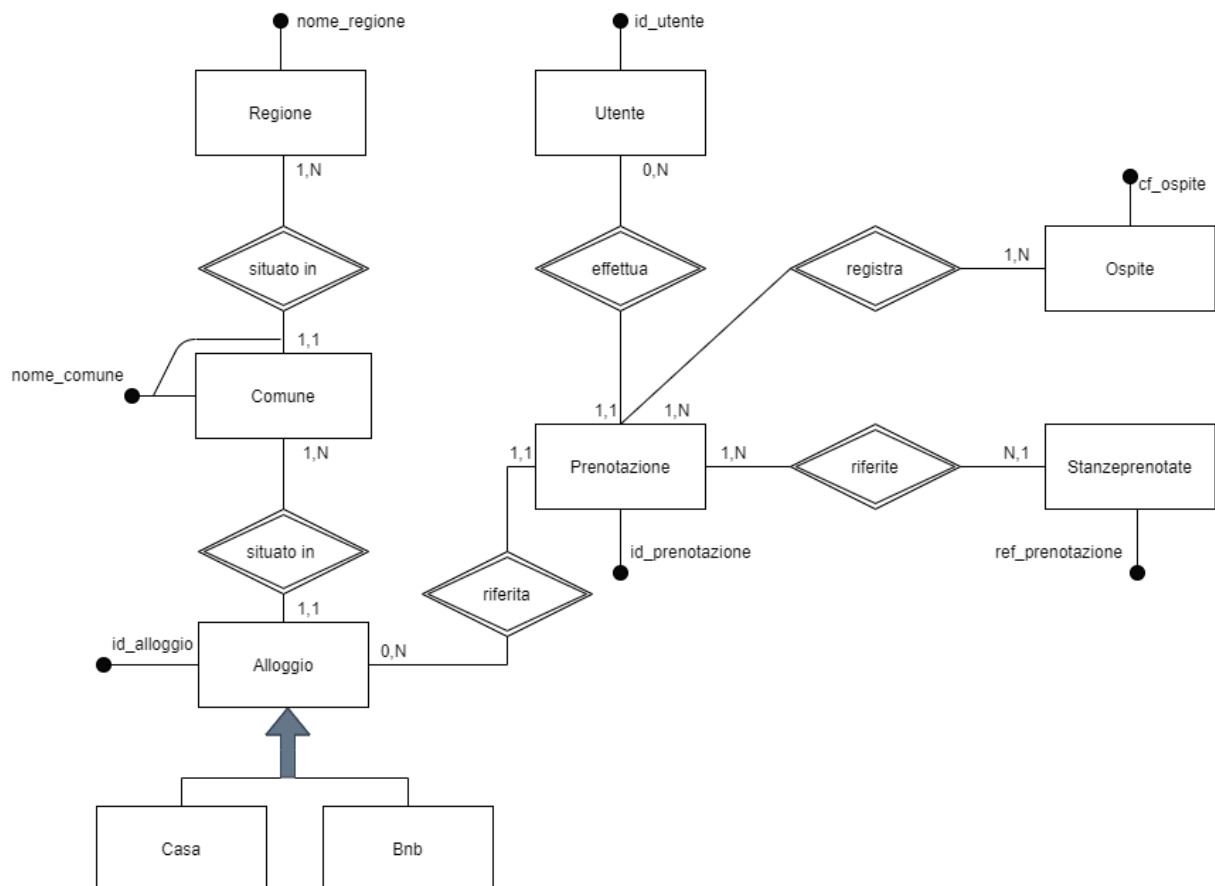


Funzionalità Admin

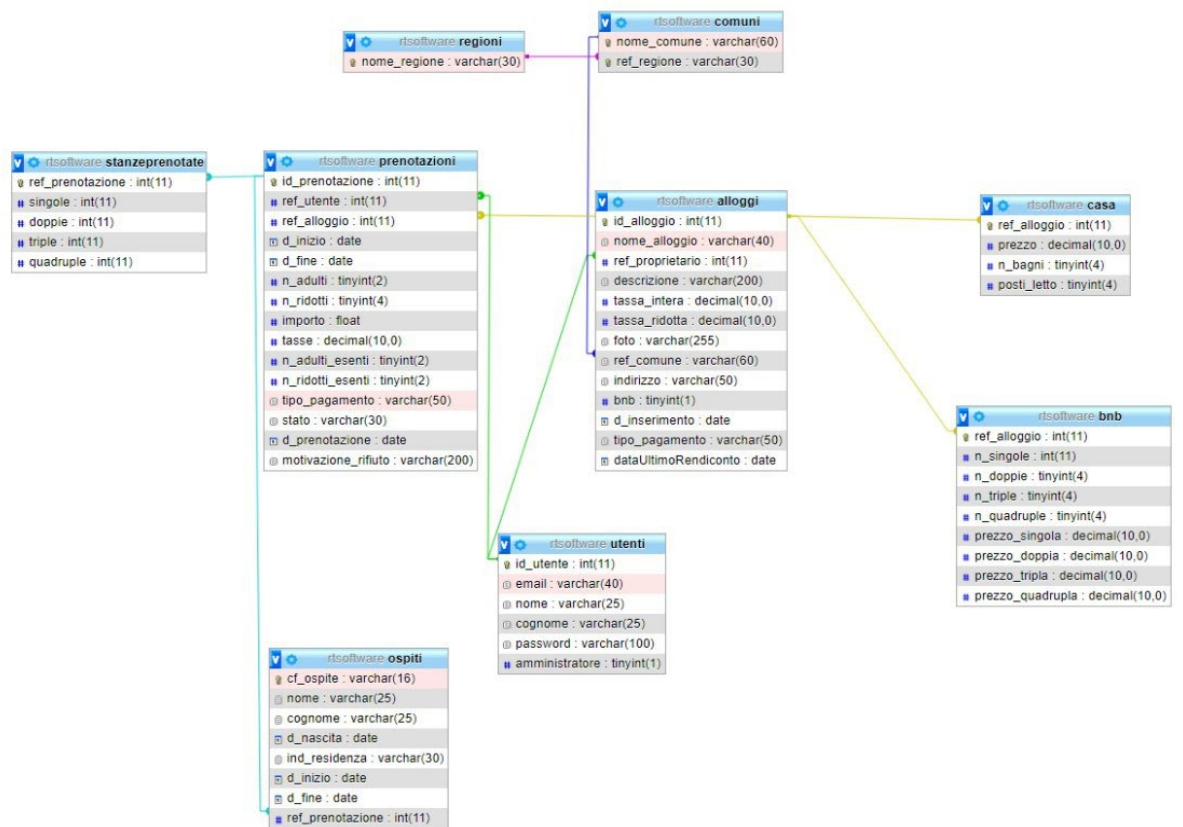


Gestione Prenotazioni





3.2.2 Modello relazionale



3.2.3 Considerazioni sul Database

Dallo schema qui sopra riportato si possono vedere i vincoli di chiavi esterne tra le diverse tabelle:

stanzePrenotate.ref_prenotazione(FK) → prenotazioni.id_prenotazione(PK)
ospiti.ref_prenotazione(FK) → prenotazioni.id_prenotazione(PK)
prenotazioni.ref_utente(FK) → utenti.id_utente(PK)
prenotazioni.ref_alloggio(FK) → alloggi.id_alloggio(PK)
comuni.ref_regione(FK) → regioni.nome_regione(PK)
alloggi.ref_comune(FK) → comuni.nome_comune(PK)
alloggi.ref_proprietario(FK) → utenti.id_utente(PK)
bnb.ref_alloggio(FK) → alloggi.id_alloggio(PK)
casa.ref_alloggio(FK) → alloggi.id_alloggio(PK)

Alcuni attributi sono vincolati ad un particolare dominio di valori, questi vincoli sono descritti, ove esistenti, nelle tabelle qui di seguito.

3.2.4 Descrizione delle tabelle

Utenti

Nome Colonna	Tipo	Vincoli	Descrizione
id_utente	Int	A.I. , PK, NOT NULL	Identificativo univoco dell'utente
nome	Varchar(25)	Not null	Nome dell'utente
cognome	Varchar(25)	Not null	Cognome dell'utente
email	Varchar(40)	Not null	Indirizzo email associato all'utente
password	Varchar(100)	Not null	Password (cifrata)
amministratore	Boolean	Not null	Indica se l'utente è amministratore (1) o meno (0)

Alloggi

Nome Colonna	Tipo	Vincoli	Descrizione
id_alloggio	Int	A.I. , PK, NOT NULL	Identificativo dell'alloggio
nome_alloggio	Varchar(30)	NOT NULL	Nome dell'alloggio
ref_proprietario	Int	FK, Not null	Codice identificativo del proprietario dell'alloggio
descrizione	Varchar(200)		Descrizione inserita dell'utente relativa all'alloggio
tassa_intera	Float	Not null	Tassa da far pagare per chi ha un'età maggiore di anni 12
tassa_ridotta	Float	Not null	Tassa da far pagare per chi ha un'età minore/uguale di anni 14
bnb	boolean	Not null	Tipologia di alloggio: 0=casa vacanze; 1= B&B
ref_comune	Int	Not null	Città dove si trova l'alloggio

indirizzo	Varchar(30)	Not null	Indirizzo di dove si trova l'alloggio
foto	Varchar(255)		Foto relativa all'alloggio
d_inserimento	Date	Not null	Data di inserimento dell'alloggio
tipo_pagamento	Varchar(50)	Not null	I tipi di pagamento consentiti per l'alloggio: online, contanti, entrambi
dataUltimoRendi conto	Date		Data nella quale si è eseguito per l'ultima volta il rendiconto

Ospiti

Nome Colonna	Tipo	Vincoli	Descrizione
cf_ospite	Varchar(16)	PK	Codice fiscale dell'ospite di una struttura
d_nascita	Date	Not null	Data di nascita dell'ospite
ind_residenza	Varchar(30)	Not null	Indirizzo di residenza dell'ospite
nome	Varchar(25)	Not null	Nome dell'ospite
cognome	Varchar(25)	Not null	Cognome dell'ospite
d_inizio	Date	Not null	Data inizio permanenza nella struttura
d_fine	Date	Not null	Data fine permanenza nella struttura
ref_prenotazione	Int	Not null	Riferimento alla prenotazione

Prenotazioni

Nome Colonna	Tipo	Vincoli	Descrizione
id_prenotazione	Int	A.I. , PK, Not null	Identificativo univoco della prenotazione
ref_utente	Int	FK, Not null	Identificativo dell'utente che ha effettuato la prenotazione
ref_alloggio	Int	FK, Not null	Identificativo dell'alloggio
importo	Float	Not null	Importo dovuto, al netto delle tasse di soggiorno
tasse	Float	Not null	Totale da pagare relativo alle tasse di soggiorno
d_inizio	Date	Not null	Data di inizio soggiorno
d_fine	Date	Not null	Data di fine soggiorno
n_adulti	Int	Not null	Numero di ospiti sopra i 12 anni
n_ridotti	Int	Not null	Numero di ospiti sotto i 12 anni
stato	Varchar(30)	Not null	Indica lo stato della prenotazione (confermata, in attesa, in corso)

n_adulti_esenti	Int	Not null	Numero di ospiti con età maggiore di 12 anni, esentati dal pagamento della tassa di soggiorno
n_ridotti_esenti	Int	Not null	Numero di ospiti con età minore/uguale di 12 anni, esentati dal pagamento della tassa di soggiorno
tipo_pagamento	Varchar(50)	Not null	Tipo di pagamento con cui si è deciso di pagare: online, contanti
d_prenotazione	Date	Not null	Data inserimento della richiesta di prenotazione
motivazione_rifiuto	Varchar(200)		Motivazione del rifiuto/annullamento

StanzePrenotate

ref_prenotazione	int	PK, Not null	Chiave esterna alla tabella prenotazione
singole	int		numero di singole che si vorrebbero prenotare
doppie	int		numero di doppie che si vorrebbero prenotare
triple	int		numero di triple che si vorrebbero prenotare
quadruple	int		numero di quadruple che si vorrebbero prenotare

Regioni

nome_regione	Varchar(30)	PK, Not null	Nome della regione
--------------	-------------	--------------	--------------------

Comuni

nome_comune	Varchar(60)	PK, Not null	Nome del comune
ref_regione	Varchar(30)	PK, FK, Not null	Nome della

			regione nella quale il comune si trova
--	--	--	--

Casa

ref_alloggio	Int	PK, Not null	Riferimento alla tabella alloggio
prezzo	Float	Not null	Prezzo per una notte
n_bagni	Int	Not null	Numero bagni disponibili
posti_letto	Int	Not null	Numero posti letto disponibili

BnB

ref_alloggio	Int	PK, Not null	Riferimento alla tabella alloggio
n_singole	Int	Not null	Numero singole disponibili
n_doppie	Int	Not null	Numero doppie disponibili
n_triple	Int	Not null	Numero triple disponibili
n_quadruple	Int	Not null	Numero quadruple disponibili
prezzo_singola	Float	Not null	Prezzo singola per una notte
prezzo_doppia	Float	Not null	Prezzo doppia per una notte

prezzo_tripla	Float	Not null	Prezzo tripla per una notte
prezzo_quadrupla	Float	Not null	Prezzo quadrupla per una notte

3.3 Accesso al sistema e sicurezza

Per l'accesso al sistema non è necessario il riconoscimento mediante credenziali.

Le funzionalità di registrazione, recupero password, accesso e ricerca di alloggi sono accessibili a tutte le tipologie di utente, registrato e non.

Le altre funzionalità necessitano invece della registrazione e dell'inserimento delle credenziali nell' opportuno pannelloLogin. In base alla tipologia di utente si avranno a disposizione funzionalità diverse.

3.3.1 Controlli di Integrità e Sicurezza del Sistema

- Il sistema NON memorizza le password in chiaro. La cifratura delle password permette di rendere più onerosi, se non impossibili, meccanismi per risalire alla password originaria (ad esempio: bruteforce, attacchi a dizionario, attacchi che utilizzano rainbow tables ecc...).
- Affinché si mantenga un certo livello di sicurezza, il sistema utilizzerà connessioni HTTPS, implementando quindi il Secure Socket Layer.