

**RBS**

**Reti Booking System**

**Sistema di prenotazione delle sale**

**Progetto finale – percorso circolare sviluppo**

Realizzato da: **Andrea Guzzo**

Sommario

[Introduzione 3](#_Toc18168459)

[Funzionalità 4](#_Toc18168460)

[Tecnologie utilizzate e dettagli implementativi 5](#_Toc18168461)

[Descrizione utilizzo 7](#_Toc18168462)

[Home page 7](#_Toc18168463)

[Prenotazione 7](#_Toc18168464)

[Gestisci 10](#_Toc18168465)

[Informazioni 13](#_Toc18168466)

[UX Design 15](#_Toc18168467)

[Sviluppi Futuri 16](#_Toc18168468)

[Login utente 16](#_Toc18168469)

[Richiesta di supporto 16](#_Toc18168470)

[Istruzioni di Deploy 18](#_Toc18168471)

# Introduzione

RBS o anche chiamato: Reti Booking System è un sistema di prenotazione delle sale riunioni, aule o stanze all’interno di un campus virtuale composto da diversi edifici e diverse sale.

Il progetto è stato realizzato come una soluzione full-stack web utilizzando lo stack Microsoft con .Net Framework 4.7, Sql Server, Entity Framework e Typescript.

Questa soluzione è stata costruita come esame finale del percorso circolare sviluppo Reti.

L’obiettivo di questo documento è dare una breve panoramica sul progetto realizzato.

Il codice del progetto è scaricabile anche dalla seguente repository GitHub:

<https://github.com/JeyDi/RBS>

Per qualsiasi altra informazione relativa alle richieste di esame, si fa riferimento al documento di presentazione del progetto sempre all’interno del progetto.

# Funzionalità

Il programma consente all’utente di utilizzare le seguenti funzionalità:

* Creazione di una risorsa
  + È possibile censire all’interno di un sistema una nuova risorsa inserendo il nome e cognome della persona stessa.
  + La creazione dello username e della email avviene in automatico, con la stessa logica implementata all’interno dei sistemi attuali Reti.
  + Vengono gestiti situazioni e casi di omonimia
* Elenco delle risorse e dettaglio di una risorsa
  + È possibile all’interno della tabella avere informazioni riguardo all’utente
* Creazione di un edificio
  + Un edificio è un’entità con all’interno delle sale disponibili. Possono infatti esistere più edifici all’interno del complesso virtuale rappresentato.
* Elenco degli edifici e dettaglio di un edificio
* Creazione di una sala
  + Una sala riunioni o un’aula è legata strettamente all’edificio di cui fa parte, è possibile avere più sale riunioni all’interno di ogni edificio. Inoltre ogni sala è contraddistinta da un numero di posti disponibili
* Elenco delle sale e dettaglio di una sala
* Inserimento di una prenotazione
  + Il cuore del sistema è la possibilità di creare delle prenotazioni, una prenotazione è costituita da diverse caratteristiche quale la data e l’ora di una prenotazione, la descrizione, l’evento, ma soprattutto è legata strettamente ad una stanza e ad un utente.
  + Non è possibile quindi creare delle prenotazioni “orfane” svincolate dalla stanza di riferimento oppure dall’utente che ha effettuato una nuova prenotazione.
* Elenco delle prenotazioni ed eliminazione di una prenotazione
  + All’interno di una tabella specifica saranno visualizzabili i dettagli della prenotazione, sarà possibile filtrare i campi ed effettuare l’eliminazione della prenotazione.

# Tecnologie utilizzate e dettagli implementativi

Le tecnologie utilizzate all’interno del progetto sono le seguenti:

* .Net Framework 4.7
* C# 7
* Entity Framework
* SQL Server
* Typescript 3.5
* HTML 5
* CSS 3
* Docker

Inoltre sono state implementate e utilizzate

* CORS (Cross-origin resource sharing)
* Swagger
* NuGet e NPM come package managers
* Librerie di terze parti per la realizzazione del front-end:
  + JQuery
  + Bootstrap
  + Bootstrap-table
  + Bootstrap-datepicker
  + Fontawesome

Sono stati utilizzati e implementati i seguenti pattern applicativi:

* Repository
* Unit of work

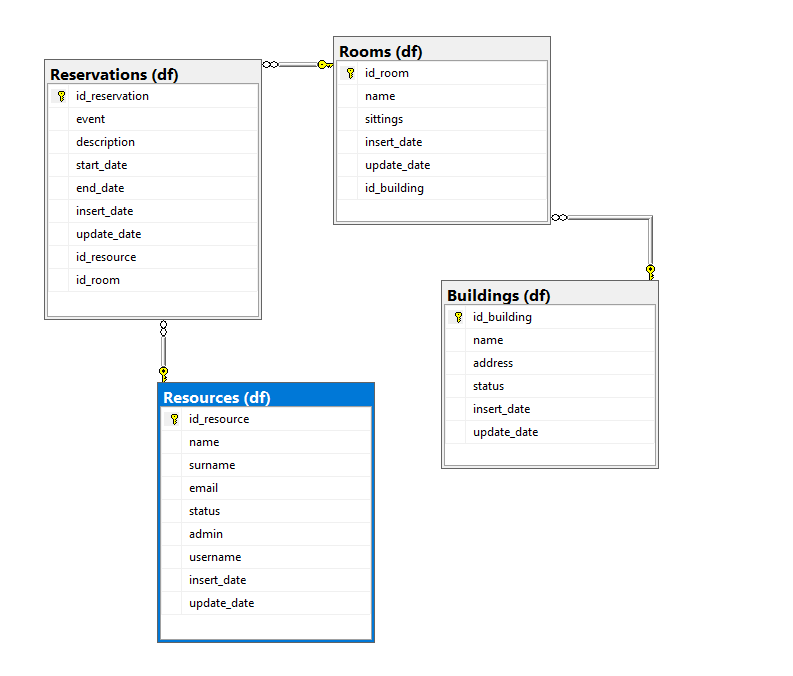
Per l’utilizzo di Entity Framework è stato adottato un approccio Database first in quanto buona parte delle logiche applicative risiede all’interno delle stored procedure.

Le procedure sono state quindi mappate correttamente all’interno del backend estendendone, qual ora fosse necessario, le funzionalità.

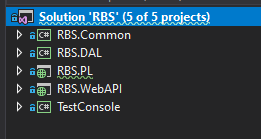
Questa scelta è stata perché si è voluto simulare in questo progetto un caso realistico di un’applicazione complessa Enterprise. Certamente un approccio code-first avrebbe favorito e agevolato lo sviluppo, ma è stato preferito questo approccio anche ai fini di sperimentazione e studio.

Il Database è stato realizzato utilizzando un’immagine ufficiale Microsoft Docker per Linux

Viene riportato inoltre lo schema del database



Il progetto inoltre è stato suddiviso nelle seguenti entità:



All’interno del progetto: Common è possibile trovare la definizione di tutti gli oggetti di business e di logiche aggiuntive di business comuni a tutto il progetto.

DAL invece contiene la definizione del Data Access Layer, ovvero di Entity Framework, dei patterns e il mapping di stored procedure e tabelle.

PL (Presentation Layer) contiene le pagine web e tutto il progetto lato front-end

WebAPI invece contiene il progetto di web api con i controllers realizzati.

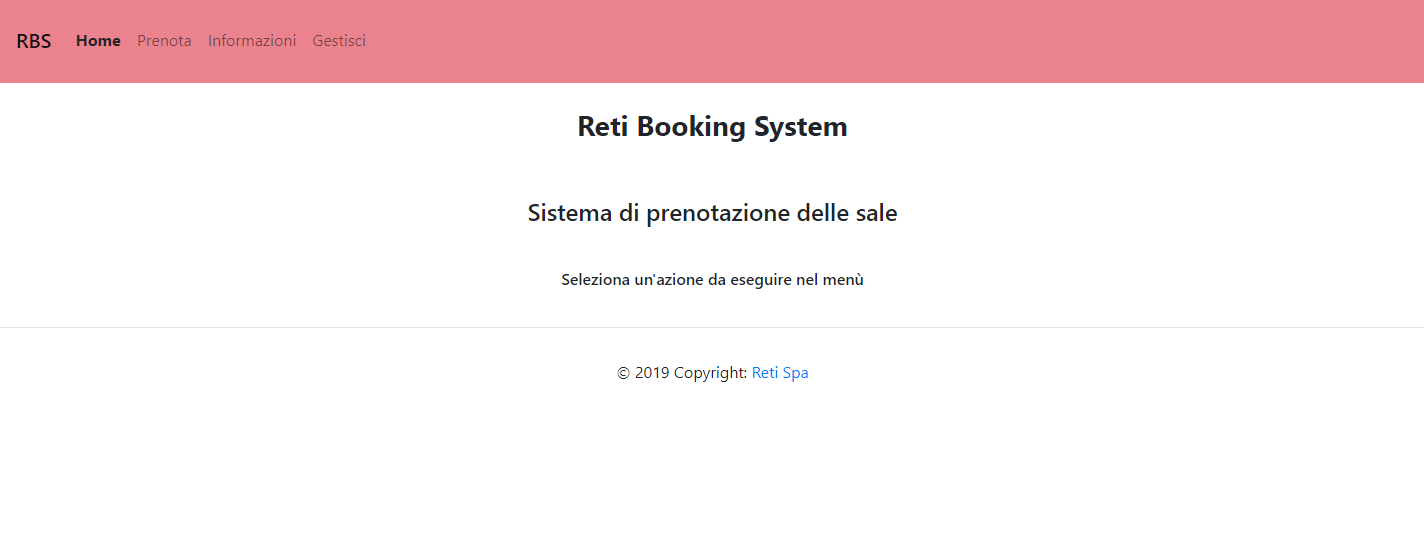
Test Console infine è una console di test che è stata utilizzata per effettuare alcune prove in fase di sviluppo.

# Descrizione utilizzo

In questo capitolo vengono illustrate le funzionalità del progetto

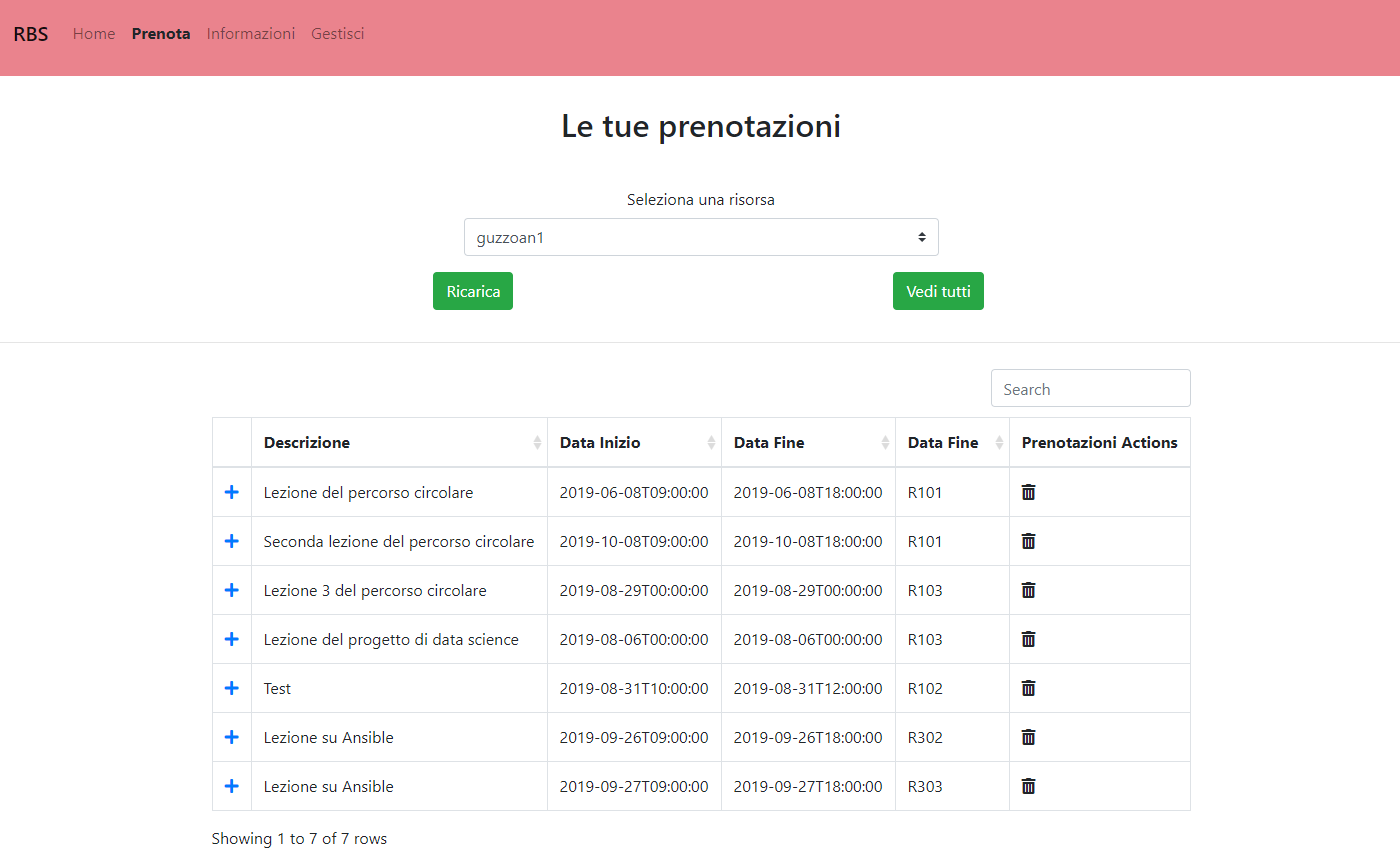
## Home page

Lanciando il progetto compare una schermata di benvenuto, è possibile interagire con le varie funzionalità utilizzando l’apposito menù in alto nella pagina



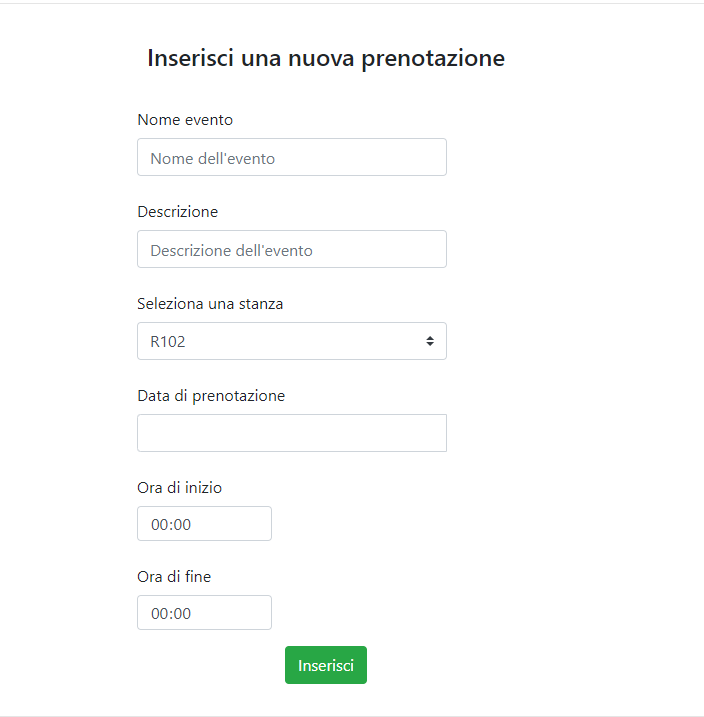
## Prenotazione

All’interno del menù di prenotazione sarà possibile visualizzare tutte le prenotazioni di un singolo utente selezionandolo nella rispettiva dropdown, oppure visualizzare tutte le prenotazioni fatte all’interno del sistema



Premendo sul pulsante + colorato di blu all’interno della tabella è possibile visualizzare i dettagli delle prenotazioni, mentre premendo sull’icona del cestino è possibile eliminare la prenotazione effettuata.

Sotto alla tabella è disponibile il form di inserimento di una nuova prenotazione, compilando i relativi campi è possibile inserire una prenotazione per l’utente specificato in alto nella pagina.



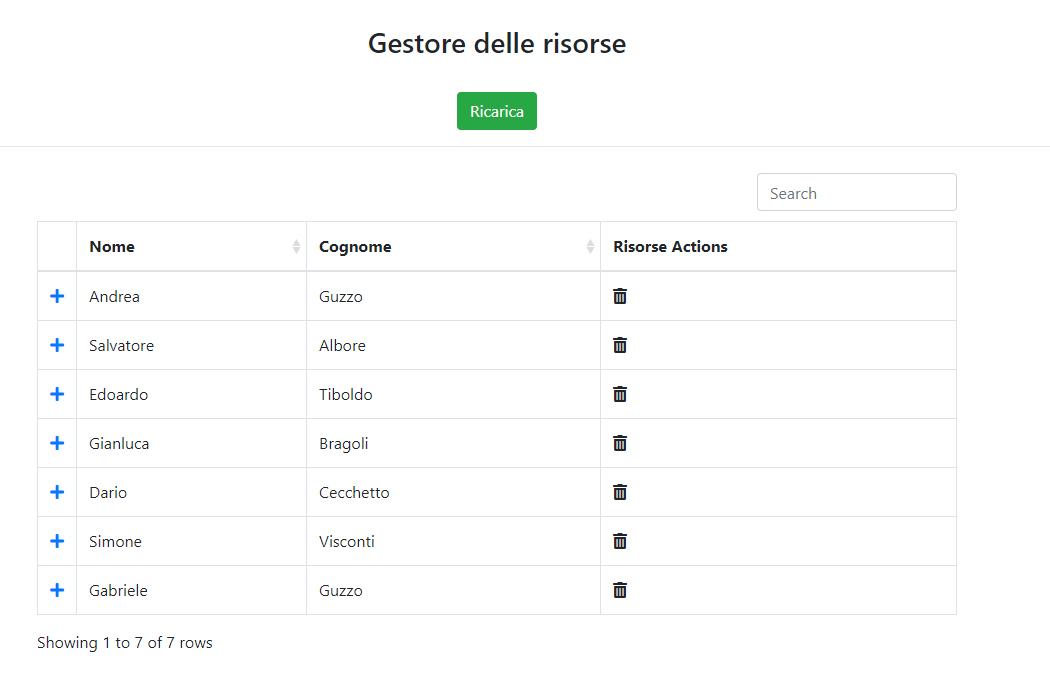
Nota: è possibile interagire con il componente della data di prenotazione premendo sulla rispettiva data nel calendario a comparsa. È possibile effettuare prenotazioni singole su uno stesso giorno, non è possibile effettuare una prenotazione dilazionata su più giorni. Per fare ciò è necessario inserire più prenotazioni, una per ogni giorno.

## Gestisci

La pagina di gestione all’apertura avrà la seguente grafica:



Al suo interno sarà possibile premere sul rispettivo pulsante per far apparire il menù scelto relativo alle Risorse, alle Sale oppure agli Edifici.





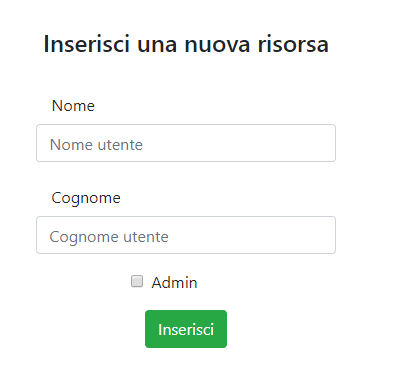


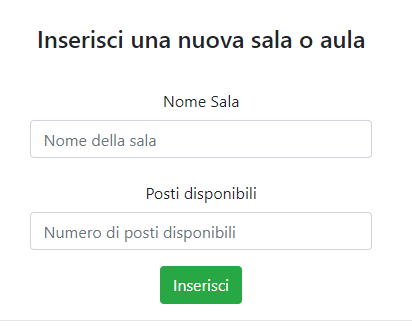
Tutti e tre i gestori sono a comparsa, premendo sul relativo pulsante in cima alla pagina verrà mostrata la sezione di riferimento.

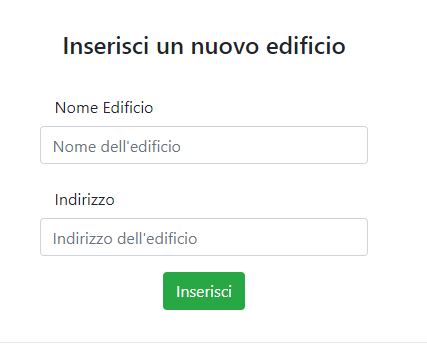
Il funzionamento dei gestori è simile per tutti e tre le entità e il funzionamento della tabella è simile a quella presentata precedentemente.

L’unica differenza è per il gestore delle sale che sarà necessario configurare selezionando correttamente l’edificio per il quale si vogliono visualizzare le rispettive sale esistenti.

Sempre al di sotto delle tabelle saranno presenti le relative schermate di input delle entità

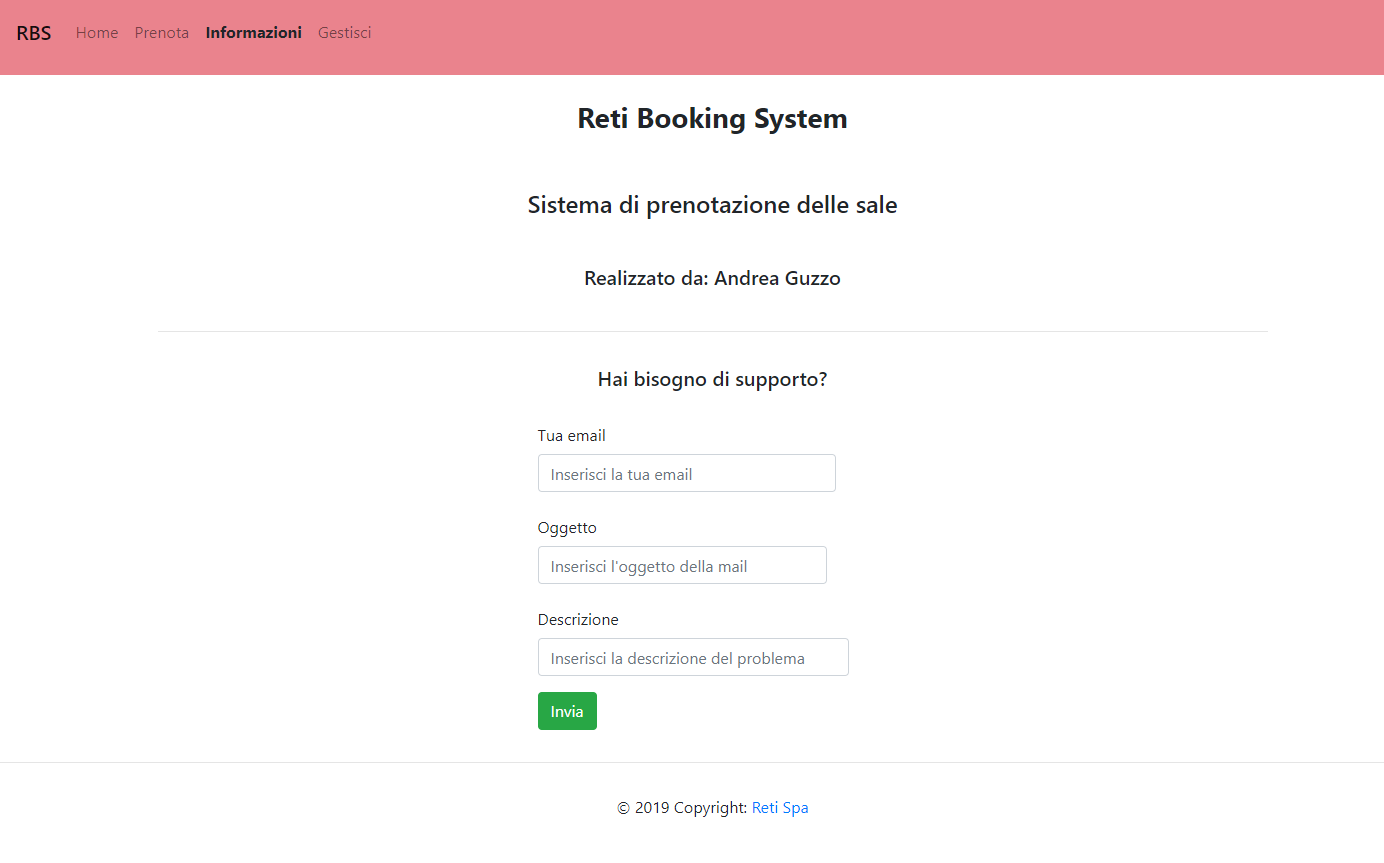






## Informazioni

La pagina delle informazioni invece contiene delle descrizioni utili e un form di supporto compilabile



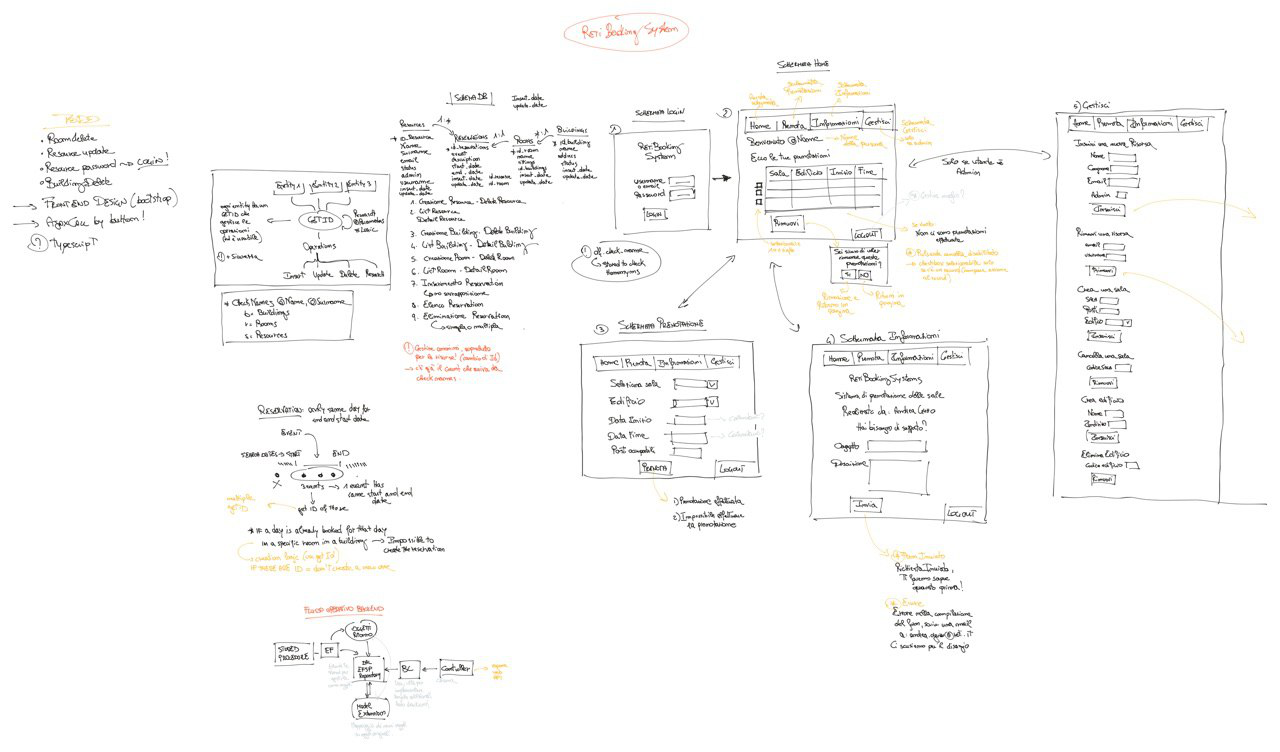
Nota: l’invio della email non è al momento funzionante a causa di problemi con il client SMTP di Yahoo Mail

ATTENZIONE: ai fini di test, oltre al progetto web, comparirà anche la schermata di Swagger che consente di testare le chiamate back-end tramite un’apposita interfaccia.

# UX Design

Per la costruzione della pagina web è sono state realizzate anche delle bozze di design della pagina in modo da renderla facilmente accessibile e utilizzabile.

Viene quindi allegato un esempio dello studio realizzato a livello grafico



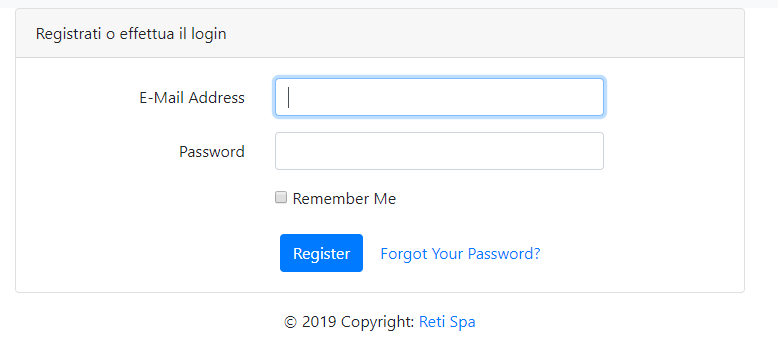
# Sviluppi Futuri

Tra i possibili sviluppi futuri ci sono due funzionalità chiave che sono già state predisposte, ma non integrate definitivamente all’interno del sistema per motivi di tempo.

## Login utente

La schermata di login è stata creata a livello grafico, ma non è stata ancora implementata.

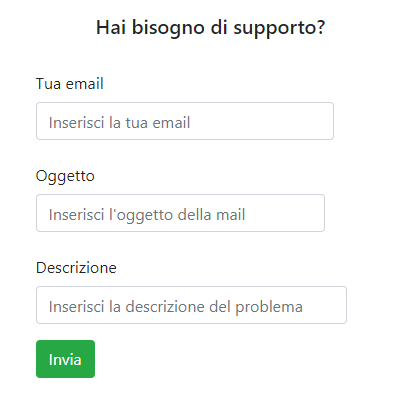
L’utilizzo della funzione di login consentirebbe di identificare in modo univoco l’utente, facilitando l’interazione e la gestione dei permessi all’interno della pagina in modo da nascondere elementi sensibili o prenotazioni private



## Richiesta di supporto

La richiesta di supporto all’interno della pagina Informazioni è stata completata e sono state realizzate anche le rispettive funzioni lato back-end.

A causa di un problema con il provider Yahoo Mail e del protocollo SMTP utilizzato non è possibile al momento effettuare una richiesta di supporto via email.



# Istruzioni di Deploy

Per il deploy è necessario scaricare da Github la repository clonandola oppure utilizzando lo Zip fornito con la consegna.

Lanciare la costruzione del database a seconda del sistema di riferimento: Linux o Windows utilizzando ed eseguendo lo script situato all’interno della cartella: db\_scripts.

Successivamente aprire la soluzione di visual studio, effettuare una build di progetto ed eventualmente andare all’interno del progetto: PL e fare un npm install in modo da installare tutte le dipendenze necessarie.

Successivamente premere il tasto: F5 ed eseguire il progetto.

Compariranno due pagine web:

* La pagina principale del progetto
* La pagina di configurazione Swagger per effettuare dei test lato backend con le WebAPI

Nota: potrebbe esser necessario aggiornare Typescript installato all’interno della macchina in quanto la versione utilizzata è la 3.5

Nota: è possibile utilizzare una versione di SQL server su una immagine Docker appositamente configurata