



**Università  
degli Studi  
di Palermo**



Università degli Studi di Palermo

Corso di laurea in Ingegneria Informativa a.a.2020/2021

Ingegneria e Progettazione del Software

Programmazione Web e Mobile

## **SYSTEM DESIGN DOCUMENT**

---



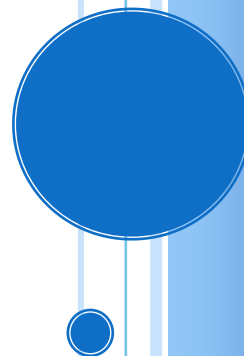
**Gruppo GAAS**

Marianna Francesca Amalfi

Vittorio Sanfilippo

Giuseppe Gullo

Michele Abanese



# **SDD**

## **(System Design Document)**

### **Indice**

- 1. Obiettivi Generali**
- 2. Obiettivi di Progettazione**
- 3. Architettura Software Proposta**
  - 3.1.**     Panoramica
  - 3.2.**     Decomposizione in sottoinsiemi
    - 3.2.1.**     Suddivisione degli oggetti all'interno dei sottoinsiemi e delle componenti
      - 3.2.1.1.** FrontEndPrincipale
      - 3.2.1.2.** Autenticazione
      - 3.2.1.3.** FrontEndCliente
      - 3.2.1.4.** FrontEndAmministratore
      - 3.2.1.5.** FrontEndAddettoParcheggio
      - 3.2.1.6.** FrontEndAutista
  - 3.3.**     **Mappatura Hardware / Software**
  - 3.4.**     **Gestione dei dati persistenti**
    - 3.4.1.**     Modello ER
    - 3.4.2.**     Modello ER Ristrutturato
    - 3.4.3.**     Descrizione delle tabelle
      - 3.4.3.1.** Metodo Pagamento
      - 3.4.3.2.** Mezzi
      - 3.4.3.3.** Prenotazioni
      - 3.4.3.4.** Utenti
  - 3.5.**     **Accesso al Sistema e Sicurezza**

## 1. Obiettivi Generali

Lo scopo del sistema proposto è quello di gestire il noleggio di mezzi di trasporto cittadino quali auto, moto, bici e monopattino.

Il software attraverso interfacce prevede la suddivisione dell'utenza in: **Cliente**, **Amministratore**, **Addetto\_Parcheggio** e **Autista**, fornendo alla specifica utenza l'accesso a funzionalità dedicate.

## 2. Obiettivi di Progettazione

- Il sistema deve essere correttamente visualizzato dalle ultime versioni stabili di Chrome, Mozilla Firefox e Microsoft Edge.
- È necessario che il sistema e il DBMS debbano essere sempre in attività, per
- garantire a qualsiasi generico Utente di accedere alle informazioni in qualsiasi momento.
- Il sistema deve controllare e gestire l'inserimento di input sintatticamente errato da parte degli Utenti.
- Il sistema deve garantire sicurezza e privacy in modo tale da garantire agli Utenti affidabilità e sicurezza dei dati sensibili e non.
- Il sistema deve garantire funzionalità minimali di sicurezza, come la crittografia per la password di un Utente Registrato.

## 3. Architettura Software Proposta

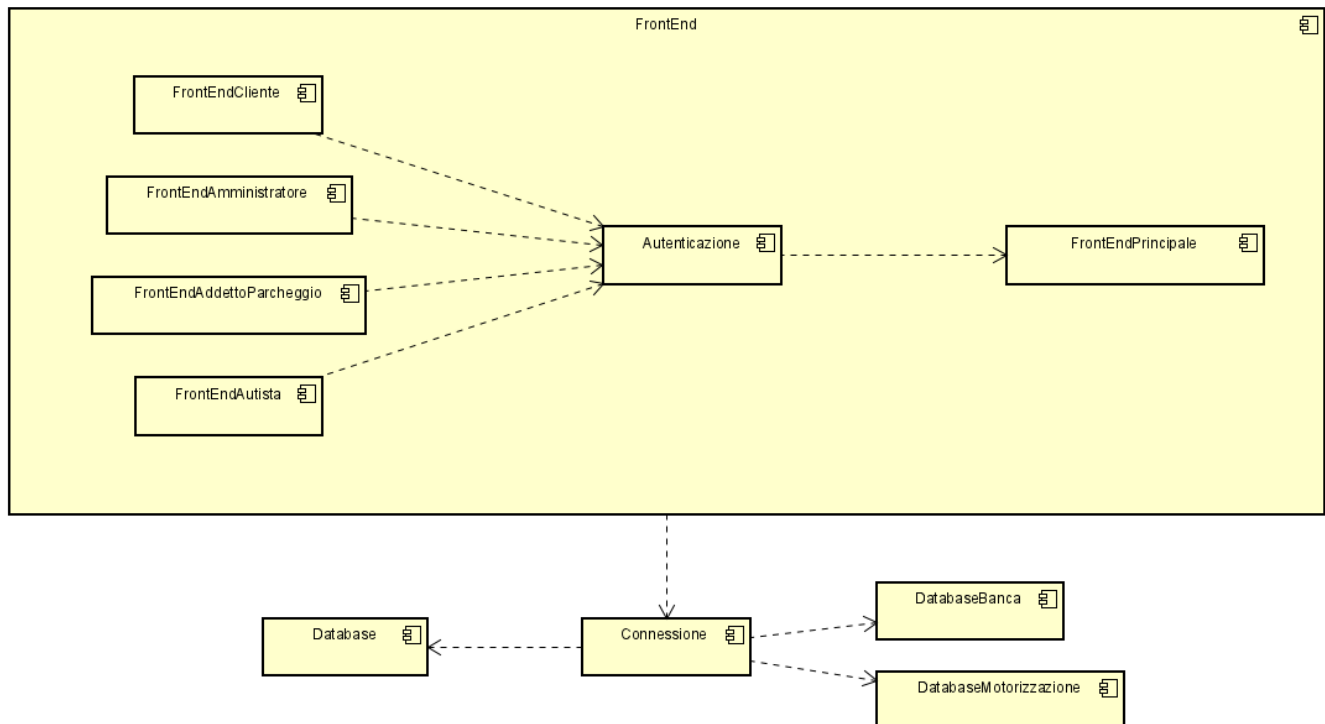
### 3.1. Panoramica

Per il sistema è stata scelta un'architettura Model-View-Controller (MVC) ibrida con Client-Server.

- **Model:** Rappresenta il nucleo del sistema, e coinvolge il sistema di gestione dei dati persistenti.
- **View:** È il componente che interagisce direttamente con l'utente, scatenando eventi e richieste. Nel sistema presentato sarà costruito mediante l'uso del framework ReactJS
- **Controller:** Questo componente si occupa di inoltrare le richieste della View al Model o in generale ai dati persistenti. Oltre che a gestire richieste rappresenta la parte logica del sistema.

L'architettura è definita ibrida per la presenza di due server. Uno di **Back-End** generato mediante Express (Server), l'altro di **Front-End** che viene generato automaticamente dal framework ReactJS (Cliente). Il secondo effettua richieste HTTPS al primo per estrapolare o elaborare dati.

### 3.2. Decomposizione in sottoinsiemi

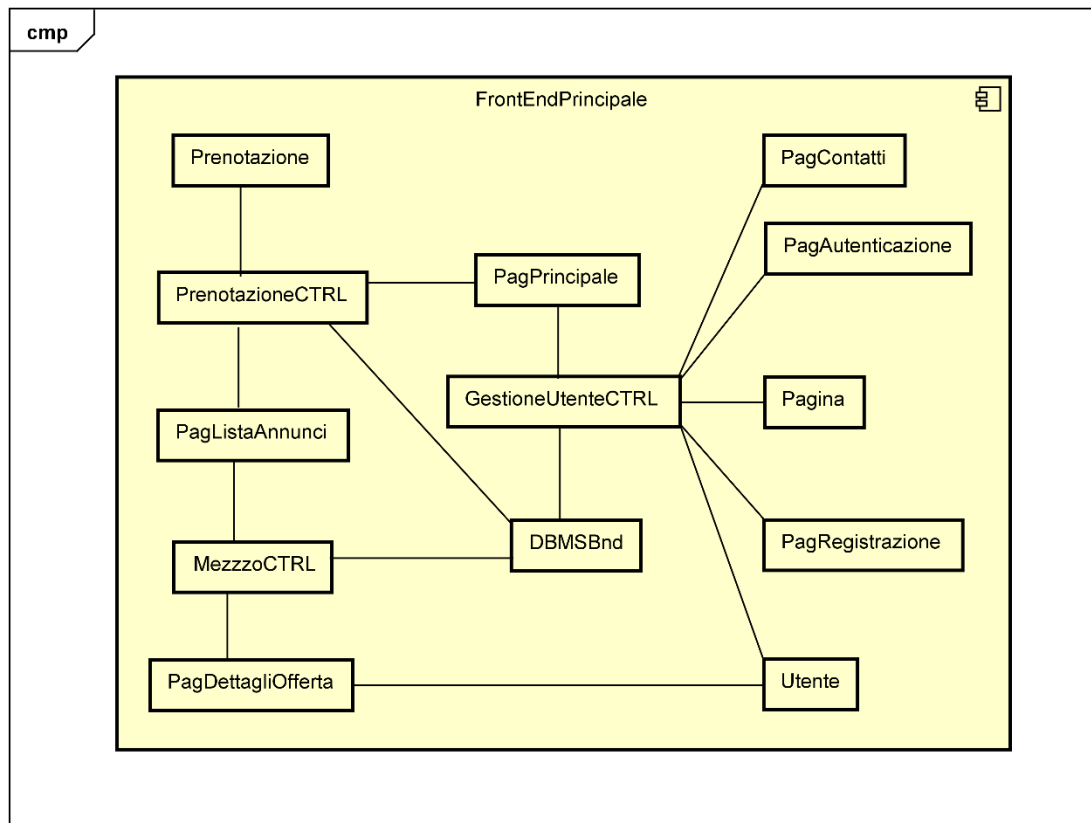


Il sottosistema **FrontEnd** racchiude i diversi sottosistemi che permettono agli utenti del software di interagire con esso. Esso risulta composto, in primo luogo, da un **FrontEndPrincipale** che racchiude le funzionalità a cui tutti possono accedere. Mediante il servizio di **Autenticazione** i diversi tipi di utenti potranno inoltre accedere a funzionalità specifiche racchiuse in **FrontEndUtenteCliente**, **FrontEndAmministratore**, **FrontEndAddettoParcheggio** e **FrontEndAutista**. Le funzionalità che ciascuno di questi ultimi sottosistemi offre sono elencate in dettaglio nei diagrammi riportati di seguito.

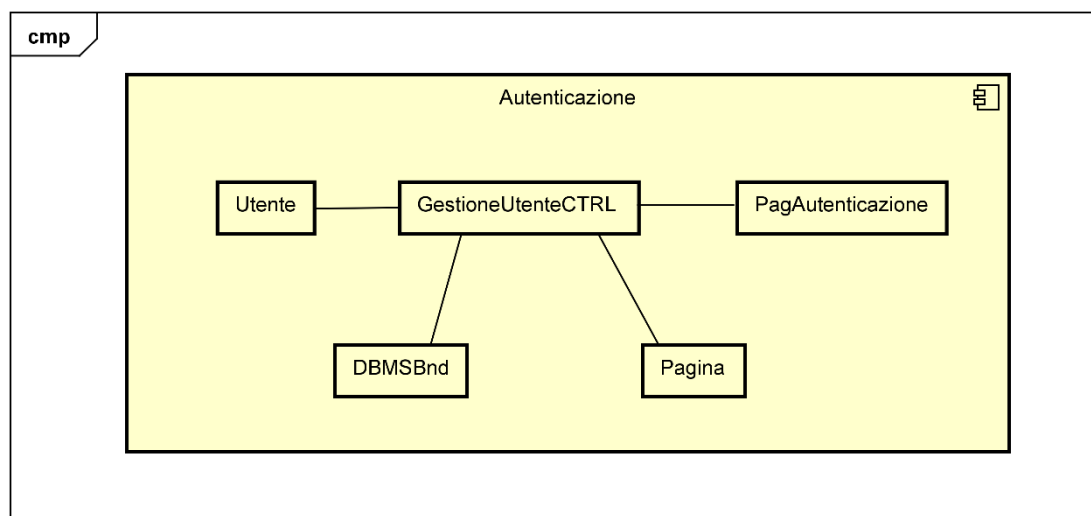
Ciascuno di questi sottosistemi ha la necessità di comunicare con il **Database**, talvolta con database esterni come il **DatabaseBanca** per la gestione del CC e le richieste di pagamento effettuate all'interno del Sistema, oppure come **DatabaseMotorizzazione** per il controllo dei dati sulla Patente inserita dall'Utente.

### 3.2.1. Suddivisione degli oggetti all'interno dei sottoinsiemi e delle componenti

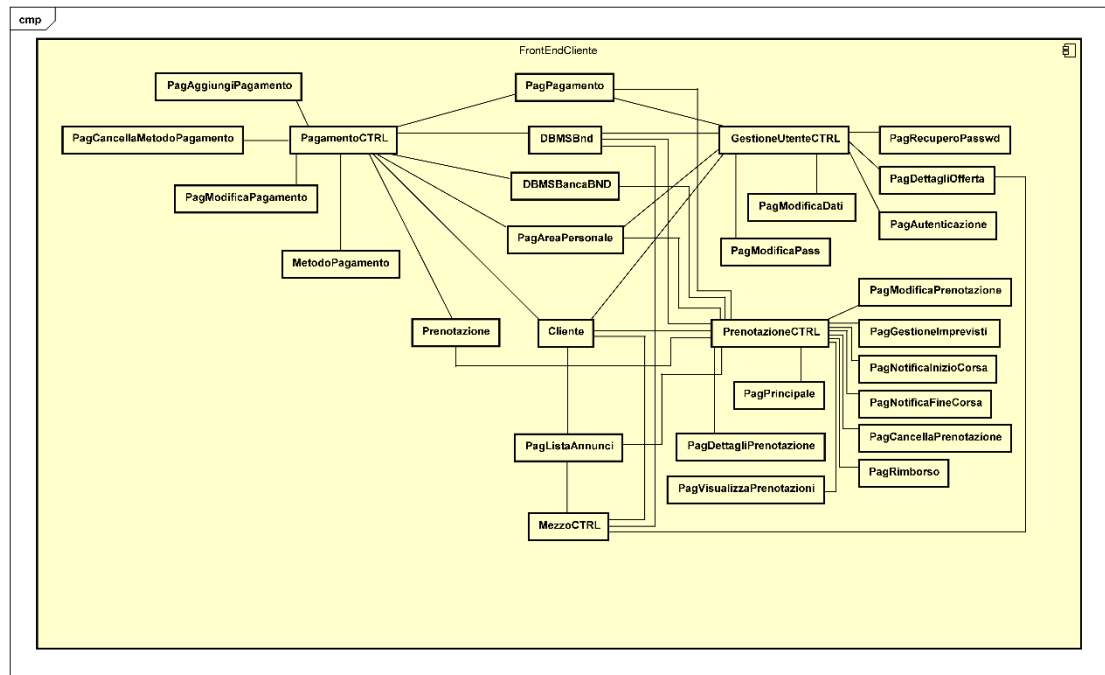
#### 3.2.1.1 FrontEndPrincipale



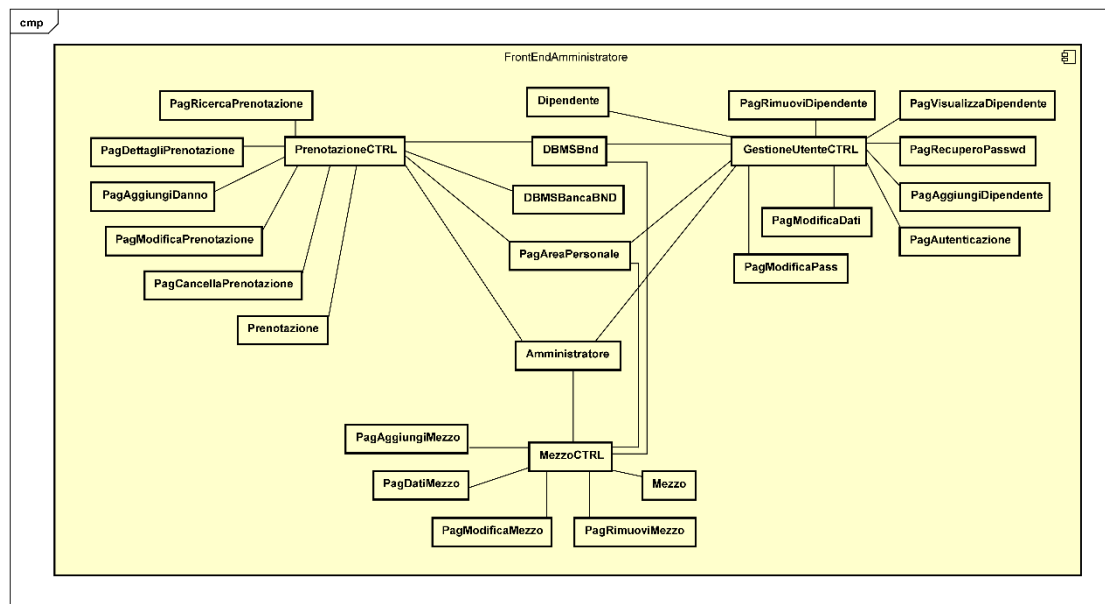
#### 3.2.1.2 Autenticazione



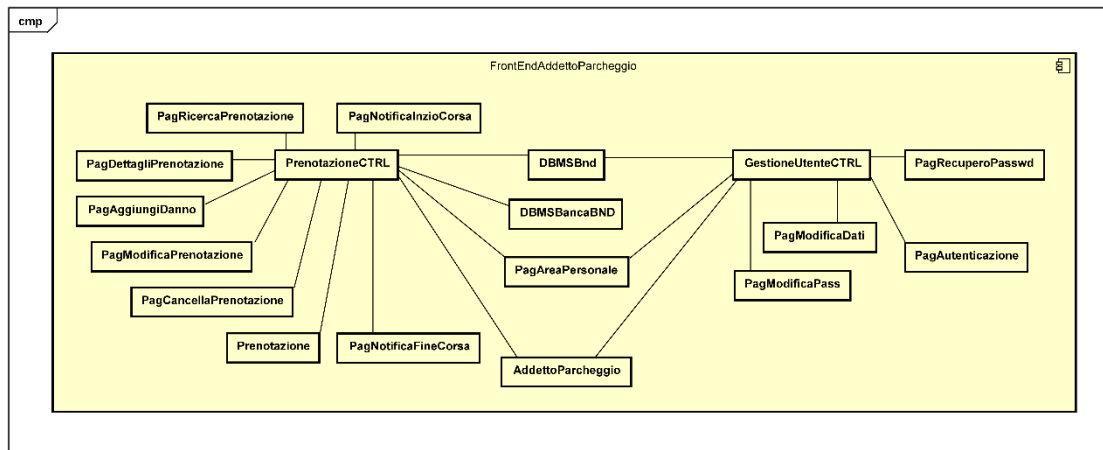
### 3.2.1.3 FrontEndCliente



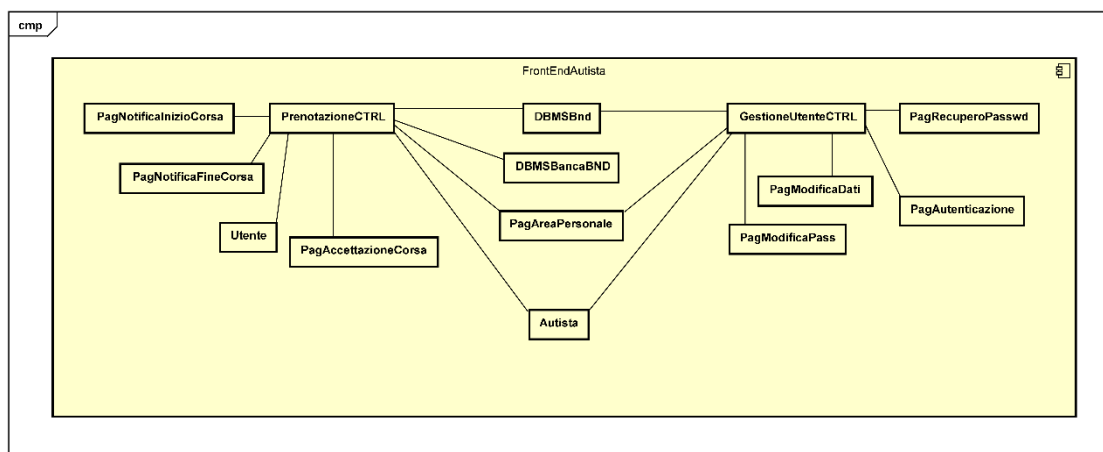
### 3.2.1.4 FrontEndAmministratore



### 3.2.1.5 FrontEndAddettoParcheggio



### 3.2.1.6 FrontEndAutista

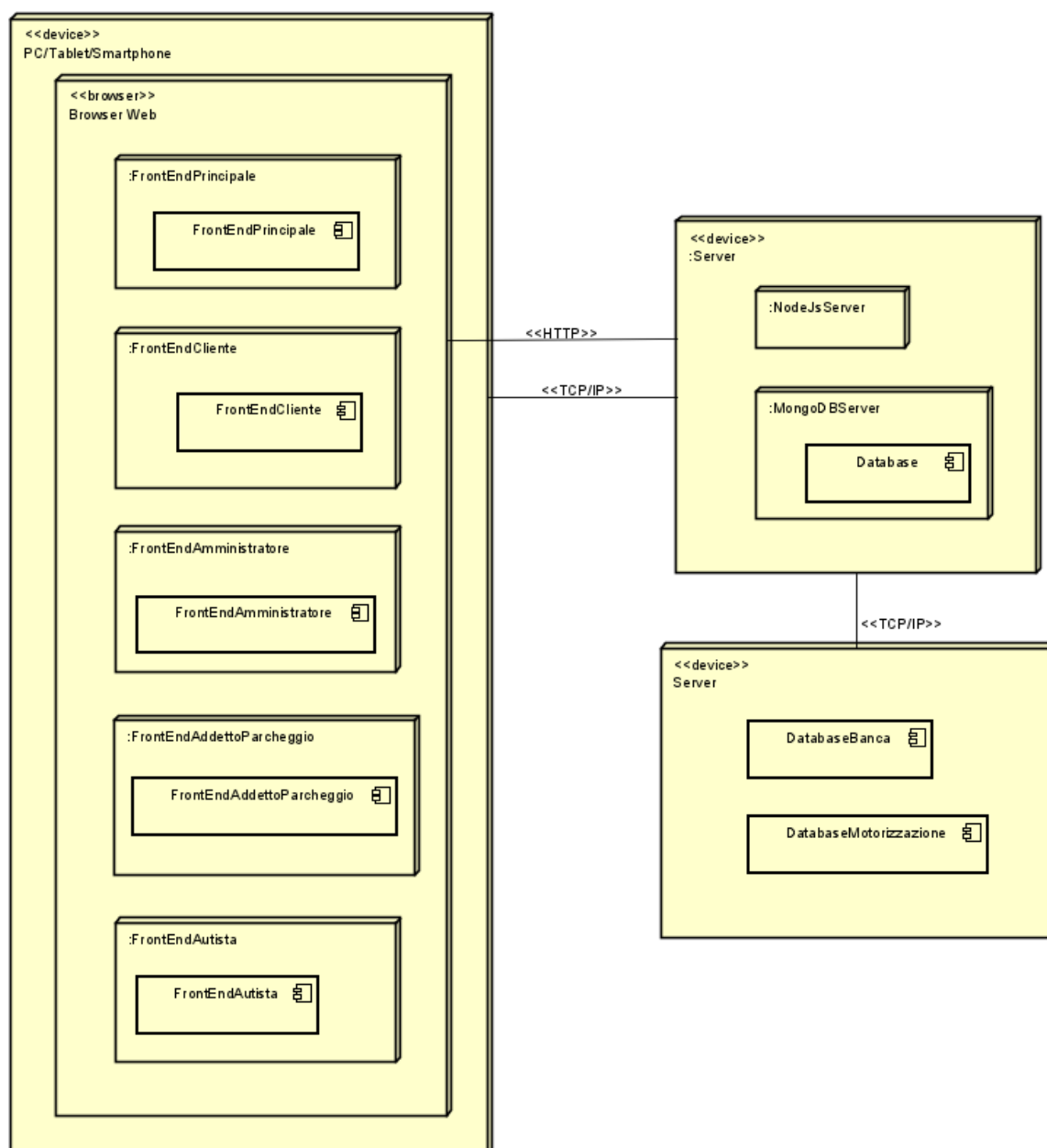


### 3.3. Mappatura Hardware / Software

La mappatura è stata effettuata sulla base del modello **Client-Server**.

Il nodo Server conterrà una istanza di **MongoDB** (Database Management System NoSQL, ovvero “Not Only SQL”, che si occuperà di gestire i contenuti del Database) e di **Node.js**, la quale interagirà con la prima al fine di soddisfare le richieste dei diversi nodi FrontEnd.

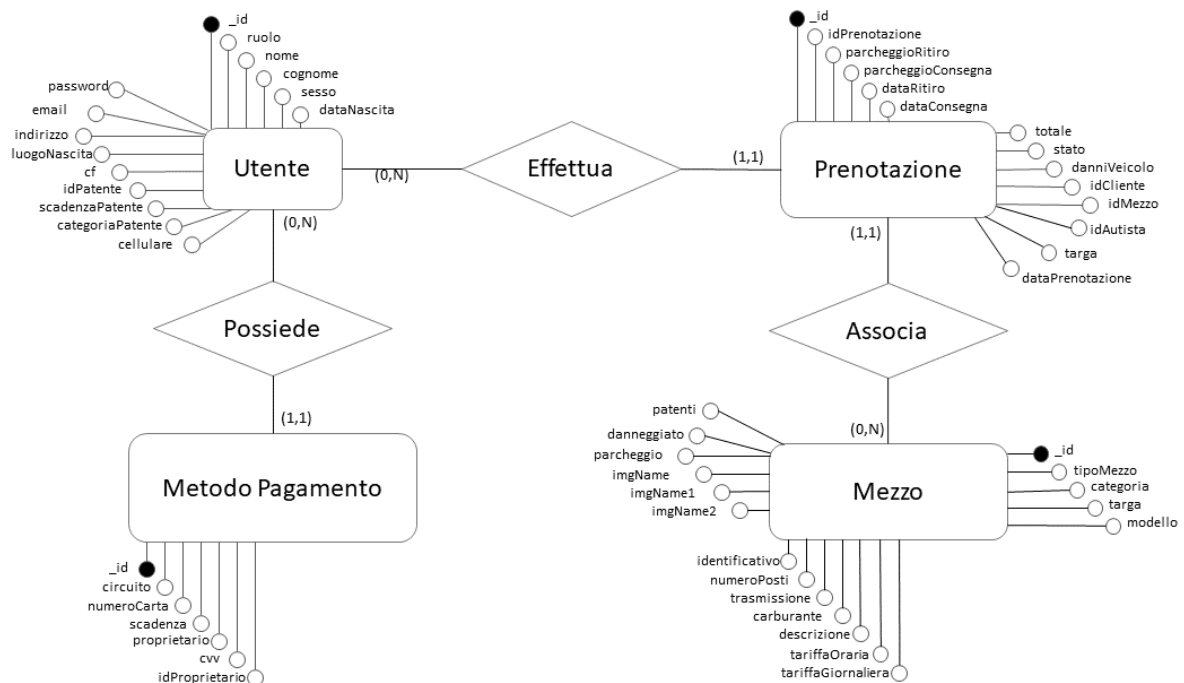
Il nodo **Client** è rappresentato da un qualunque dispositivo che sia in grado di connettersi al sistema mediante un **Browser Web**, oppure mediante applicativo in formato **Apk** in caso di connessione mediante dispositivo **Mobile Android**, attraverso il quale si potrà accedere alle funzionalità presenti nel sistema mediante i vari front-end.





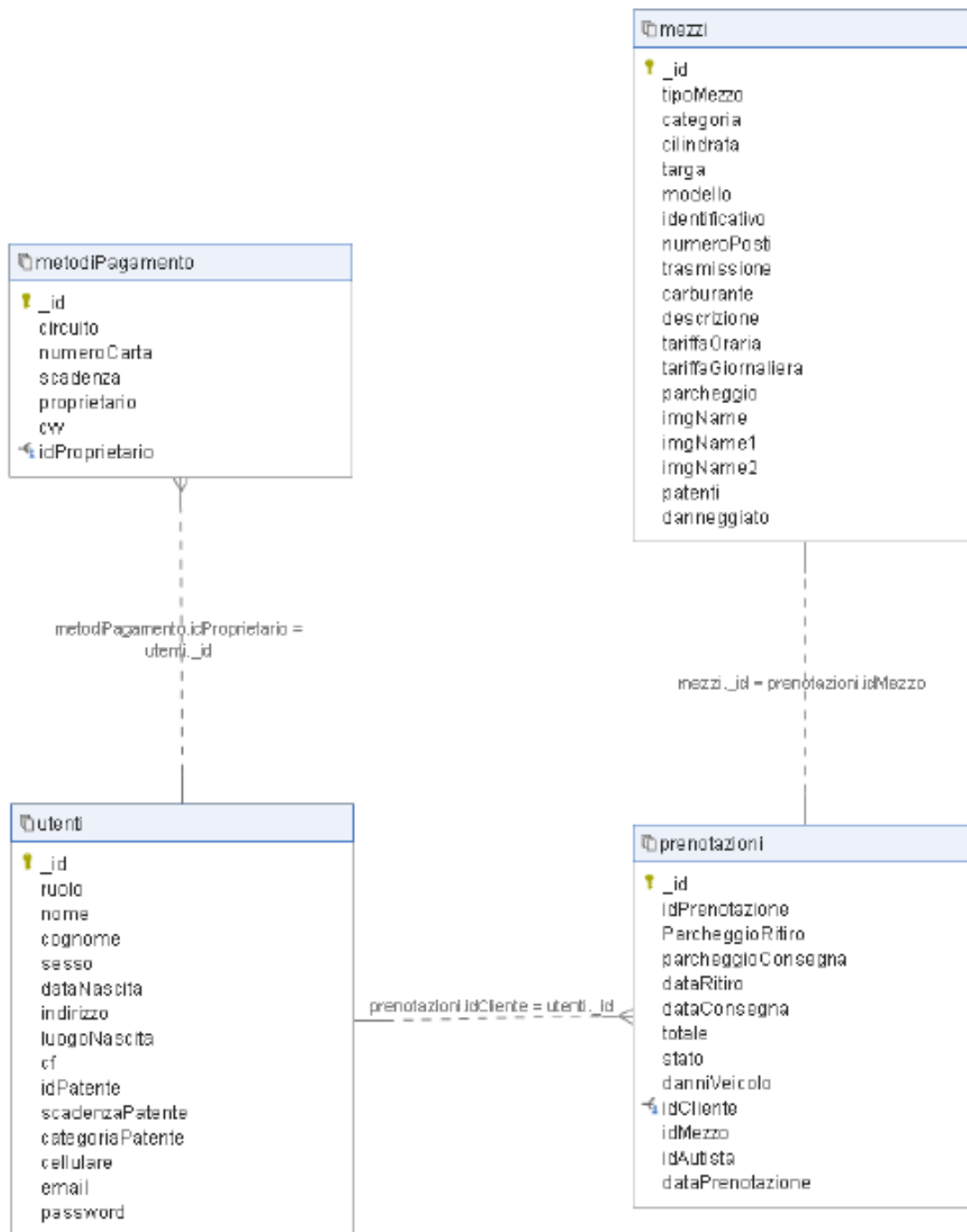
### 3.4. Gestione dei Dati Persistenti

#### 3.4.1. Modello ER



Nella rappresentazione del modello E-R sono evidenziate soltanto le chiavi primarie per facilitarne la visualizzazione.

### 3.4.2. Modello ER Ristrutturato



### 3.4.3. Descrizione delle Tabelle

#### Legend



Primary key










User-defined one to many relation




One to one relation

#### Metodo Pagamento

##### Columns

|   | Name           | Data type | Description / Attributes   |
|---|----------------|-----------|--|
|    | _id            | Id        | Identificativo univo creato dal DBMS alla creazione di una nuova tupla<br>Nullable   |
|    | circuito       | String    | Stringa che identifica la tipologia della carta<br>Nullable                          |
|    | numeroCarta    | String    | Stringa che identifica il numero della carta<br>Nullable                             |
|  | scadenza       | String    | Stringa che identifica la scadenza della carta<br>Nullable                           |
|  | proprietario   | String    | Stringa che identifica il proprietario della carta<br>Nullable                       |
|  | cvv            | String    | Stringa che identifica il Card Validation Value<br>Nullable                          |
|  | idProprietario | String    | Stringa che identifica il proprietario della carta<br>Nullable<br>References: utenti |



##### Links to

| Table  | Join   | Title / Name / Description |
|--|--|----------------------------|
|  utenti | metodiPagamentoIdProprietario =<br>utenti_id | fk_utenti_metodiPagamento  |

##### Unique keys


|   | Name / Description |
|---|--------------------|
|  _id | _id                |

##### Uses

| Name  |
|---|
|  metodiPagamento |
|  utenti          |

# Mezzi

## Columns

|   | Name       | Data type | Description / Attributes   |
|---|------------|-----------|--|
|  | _id        | Id        | Identificativo univo creato dal DBMS alla creazione di una nuova tupla<br><b>Nullable</b><br><b>References:</b> prenotazioni |
|   | tipoMezzo  | String    | Stringa che identifica la tipologia del mezzo<br><b>Nullable</b>   |
|   | categoria  | String    | Stringa che identifica la categoria del mezzo<br><b>Nullable</b>   |
|   | cilindrata | String    | Stringa che identifica la cilindrata del mezzo<br><b>Nullable</b>  |
|   | targa      | String    | Stringa che identifica la targa del mezzo<br><b>Nullable</b>   |
|   | modello    | String    | Stringa che identifica il modello del mezzo<br><b>Nullable</b>   |

|     | Name               | Data type | Description / Attributes   |
|-----|--------------------|-----------|--|
|     | identificativo     | Null      | Valore identificativo del mezzo per la gestione interna aziendale<br><b>Nullable</b>                 |
|     | numeroPosti        | String    | Stringa che identifica in numero di posti del mezzo<br><b>Nullable</b>                               |
|     | trasmissione       | String    | Stringa che identifica il tipo di trasmissione del mezzo<br><b>Nullable</b>                          |
|     | carburante         | String    | Stringa che identifica il tipo di carburante del mezzo<br><b>Nullable</b>                            |
|     | descrizione        | String    | Stringa che descrive il mezzo<br><b>Nullable</b>   |
|     | tariffaOraria      | String    | Stringa che identifica il prezzo ad ora del mezzo<br><b>Nullable</b>                                 |
|     | tariffaGiornaliera | String    | Stringa che identifica il prezzo al giorno del mezzo<br><b>Nullable</b>                              |
|     | parcheggio         | String    | Stringa che identifica il parcheggio in cui risiede il mezzo<br><b>Nullable</b>                      |
|     | imgName            | String    | Stringa che identifica un immagine del mezzo<br><b>Nullable</b>                                      |
|     | imgName1           | String    | Stringa che identifica un immagine del mezzo<br><b>Nullable</b>                                      |
|     | imgName2           | String    | Stringa che identifica un immagine del mezzo<br><b>Nullable</b>                                      |
| [ ] | patenti            | String[]  | Array che identifica le tipologie di patenti che autorizzano l'utilizzo del mezzo<br><b>Nullable</b> |
|     | danneggiato        | Boolean   | Booleano che identifica se il mezzo presenta danni o meno<br><b>Nullable</b>                         |



## Links to

| Table  | Join                           | Title / Name / Description |
|--|--------------------------------|----------------------------|
|  prenotazioni | mezzi_id = prenotazioniidMezzo | fk_prenotazioni_mezzi      |

## Unique keys


|   | Name / Description |
|---|--------------------|
|  _id | _id                |

## Uses

| Name   |
|--|
|  mezzi        |
|  prenotazioni |

# Prenotazione

## Columns

|   | Name | Data type | Description / Attributes   |
|---|------|-----------|--|
|  | _id  | Id        | Identificativo univo creato dal DBMS alla creazione di una nuova tupla<br>Nullable |

|   | Name                | Data type | Description / Attributes   |
|---|---------------------|-----------|--|
|    | idPrenotazione      | Int64     | Intero che identifica univocamente la prenotazione<br>Nullable                       |
|    | ParcheggioRitiro    | String    | Stringa che identifica il parcheggio di ritiro del mezzo<br>Nullable                 |
|    | parcheggioConsegna  | String    | Stringa che identifica il parcheggio di consegna del mezzo<br>Nullable               |
|    | dataRitiro          | DateTime  | Data che identifica l'orario di ritiro del mezzo<br>Nullable                         |
|    | dataConsegna        | DateTime  | Data che identifica l'orario di consegna del mezzo<br>Nullable                       |
|    | totale              | Int32     | Intero che identifica il prezzo totale della prenotazione<br>Nullable                |
|    | stato               | String    | Stringa che identifica lo stato del mezzo in relazione alla prenotazione<br>Nullable |
|    | danniVeicolo        | String    | Stringa che identifica la tipologia di danni al veicolo<br>Nullable                  |
|   | idCliente           | Null      | Identificativo univoco del cliente<br>Nullable<br>References: utenti                 |
|  | idMezzo             | String    | Identificativo univoco del mezzo<br>Nullable   |
|  | idMetodoPagamento   | String    | Identificativo univoco del metodo di pagamento<br>Nullable                           |
|  | idAutista           | String    | Identificativo univoco dell'autista associato alla prenotazione<br>Nullable          |
|  | targa               | String    | Stringa identificativa univoca del mezzo<br>Nullable                                 |
|  | identificativoMezzo | Null      | Identificativo univoco del mezzo<br>Nullable   |
|  | tipologia           | String    | Stringa che identifica la tipologia del mezzo<br>Nullable                            |
|  | modello             | String    | Stringa che identifica il modello del mezzo<br>Nullable                              |
|  | imgMezzo            | String    | Stringa che identifica l'immagine del mezzo<br>Nullable                              |
|  | dataPrenotazione    | String    | Stringa che identifica la data della prenotazione<br>Nullable                        |
|  | nomeCliente         | String    | Stringa che identifica il nome del Cliente<br>Nullable                               |
|  | cognomeCliente      | String    | Stringa che identifica il cognome del Cliente<br>Nullable                            |

## Links to

|   | Table  | Join                              | Title / Name / Description |
|---|--------|-----------------------------------|----------------------------|
|  | utenti | prenotazioniidCliente = utenti_id | fk_utenti_prenotazioni     |



## Linked from

|   | Table | Join                           | Title / Name / Description |
|---|-------|--------------------------------|----------------------------|
|  | mezzi | prenotazioniidMezzo = mezzi_id | fk_prenotazioni_mezzi      |



### Unique keys

|   | Name / Description |
|---|--------------------|
|  _id | _id                |

### Uses


|  | Name |
|--|------|
|  prenotazioni |      |
|  utenti       |      |

### Used By

|  | Name |
|--|------|
|  prenotazioni |      |
|  mezzi        |      |

## Utenti

### Columns

|   | Name             | Data type | Description / Attributes   |
|---|------------------|-----------|--|
|  | _id              | Id        | Identificativo univoco creato dal DBMS alla creazione di una nuova tupla<br>Nullable |
|   | ruolo            | String    | Stringa che identifica il ruolo dell'utente<br>Nullable                              |
|   | nome             | String    | Stringa che identifica il nome dell'utente<br>Nullable                               |
|   | cognome          | String    | Stringa che identifica il cognome dell'utente<br>Nullable                            |
|   | Sesso            | String    | Stringa che identifica il genere dell'utente<br>Nullable                             |
|   | dataNascita      | DateTime  | Data che identifica la data di nascita dell'utente<br>Nullable                       |
|   | indirizzo        | String    | Stringa che identifica l'indirizzo di residenza dell'utente<br>Nullable              |
|   | luogoNascita     | String    | Stringa che identifica il luogo di nascita dell'utente<br>Nullable                   |
|   | cf               | String    | Stringa che identifica il codice fiscale dell'utente<br>Nullable                     |
|   | idPatente        | String    | Identificativo univoco della patente associata all'utente<br>Nullable                |
|   | scadenzaPatente  | Null      | Data che identifica la scadenza della patente associata all'utente<br>Nullable       |
|   | categoriaPatente | String    | Stringa che identifica la tipologia della patente associata all'utente<br>Nullable   |
|   | cellulare        | String    | Stringa che identifica il numero cellulare dell'utente<br>Nullable                   |
|   | email            | String    | Stringa che identifica l'email dell'utente<br>Nullable                               |
|   | password         | String    | Stringa che identifica la password crittografata dell'utente<br>Nullable             |

Linked from

|   | Table           | Join   | Title / Name / Description |
|---|-----------------|--|----------------------------|
| → | metodiPagamento | utenti_id =<br>metodiPagamentoIdProprietario | fk_utenti_metodiPagamento  |
| → | prenotazioni    | utenti_id = prenotazioniIdCliente            | fk_utenti_prenotazioni     |

Unique keys

|       | Name / Description |
|-------|--------------------|
| 🔑 _id | _id                |

Used By

|   | Name            |
|---|-----------------|
| 📁 | utenti          |
| → | metodiPagamento |
| → | prenotazioni    |

### 3.5. Accesso al Sistema e Sicurezza

L'accesso al sistema e alla sua funzionalità base di ricerca non è vincolato in alcuna maniera, risulta quindi accessibile da parte dell'intera utenza in grado di raggiungere il sistema stesso. Mediante l'inserimento di credenziali tramite l'apposita schermata di autenticazione (previa registrazione) è possibile accedere invece alle funzionalità specifiche dedicate alla tipologia di Utente che viene ricoperta.

Per l'implementazione del sistema è stato previsto l'utilizzo delle tecnologie di cifratura delle password e di sicurezza della sessione. La cifratura viene sviluppata sfruttando la funzione di hashing 'bcrypt'. Per prevenire al meglio la session hijacking viene utilizzata la combinazione JWT + moduli express-session/session-store: il JSON Web Token è indispensabile nell'autenticazione Utente/Host e nella protezione delle rotte, mentre express-session si occupa della memorizzazione della sessione Utente, coadiuvato da session-store, il quale memorizza il payload sensibile di tale sessione nel database.