

# Università degli studi di Palermo

Dipartimento di Ingegneria  
2178 - Ingegneria Informatica  
Anno accademico 2020/21

# PickMeUp!

## System Design Document

### Docenti

*Prof.ssa* Valeria **SEIDITA** (Ingegneria del software)

*Prof.* Roberto **PIRRONE** (Programmazione web e mobile)

A cura del gruppo **Bytecoders**

# System Design Document

<b>Introduzione</b>	<b>3</b>
Obiettivi del sistema	3
Obiettivi di progettazione	3
<b>Architettura software proposta</b>	<b>4</b>
Panoramica	4
Decomposizione in sottoinsiemi	4
Mappatura degli oggetti nei sottoinsiemi	6
Autenticazione	6
Gestione Account	7
Funzionalità impiegato	8
Gestione prenotazione	10
Mappatura hardware/software	10
Progetto del database	11
Deposito	11
Veicolo	13
Prenotazione	14
Utente	16
Notifiche	18
Accesso al sistema	18

# SDD

## 1. Introduzione

### 1.1. Obiettivi del sistema

Lo scopo del sistema è quello di agevolare i compiti dell'azienda di noleggio auto, moto, bici e monopattini on demand nelle attività di gestione del servizio e l'interazione con i clienti.

### 1.2. Obiettivi di progettazione

Il sistema deve essere correttamente visualizzato dalle ultime versioni stabili dei browser Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari e Opera, sia nelle versioni desktop che mobile.

Il sistema deve impedire l'inserimento di input non validi da parte degli utenti, segnalando eventualmente il corrispettivo errore.

Inoltre, il DBMS deve sempre essere attivo per permettere agli impiegati e ai clienti di accedere al servizio software.

## 2. Architettura software proposta

### 2.1. Panoramica

L'architettura software proposta è costituita da un unico nodo centrale che contiene tutti i dati di interesse e gli utenti interagiscono con il sistema tramite i loro dispositivi personali.

Inoltre, gli utenti non hanno possibilità di interagire tra loro, quindi dovranno sempre fare richiesta al nodo centrale durante il ciclo di utilizzo del software.

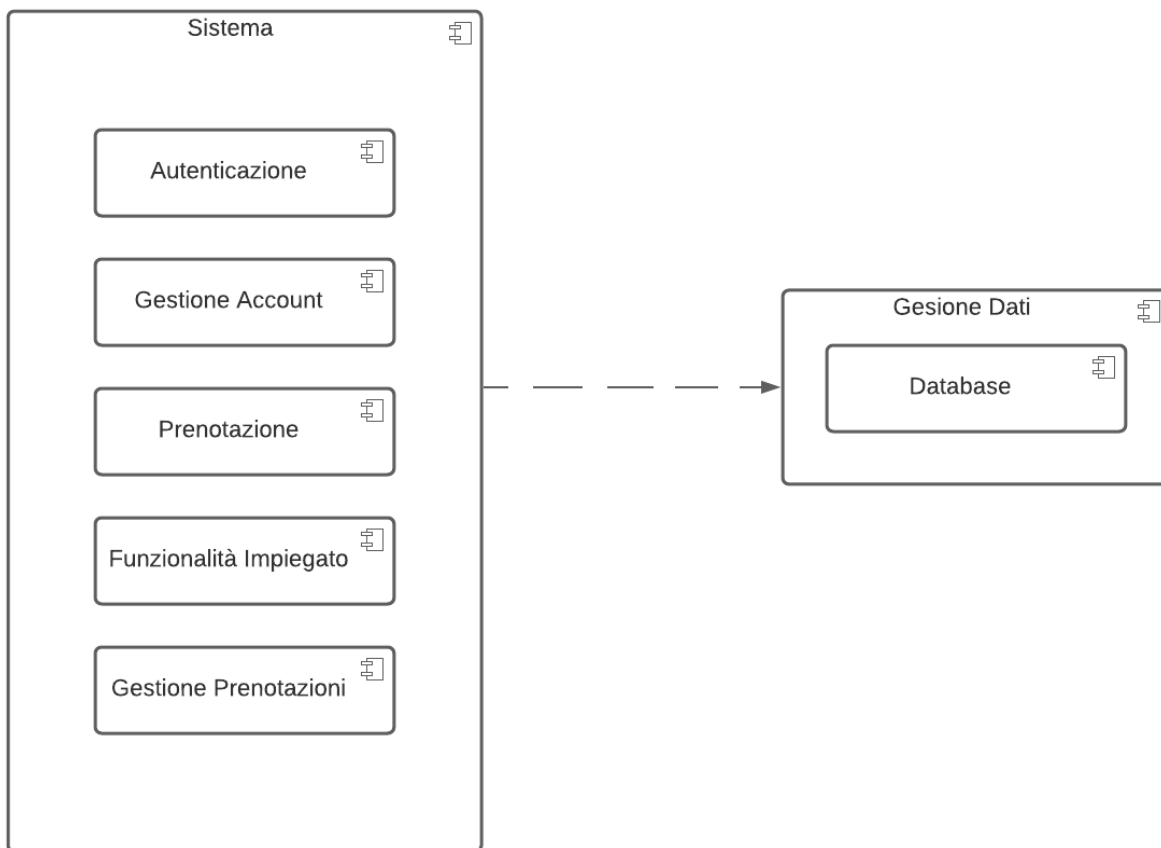
Ne deriva che il tipo di architettura più appropriata per il sistema risulta essere l'architettura Client-Server, dove i nodi client sono tutti i dispositivi che si connettono al sistema per utilizzare i servizi offerti dallo stesso.

Per la connessione e la comunicazione tra i nodi si utilizza il protocollo HTTP e, in particolare, il nodo server si occupa di rispondere alle richieste dei nodi client e della gestione del database.

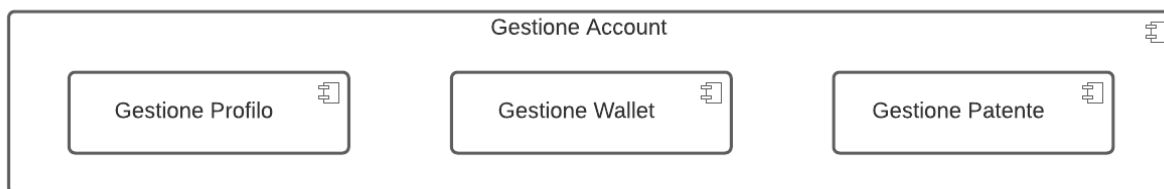
Per quest'ultimo si è scelto di utilizzare un database non relazionale orientato ai documenti JSON, realizzato tramite mongoDB, in quanto favorisce una migliore scalabilità e dinamicità del sistema.

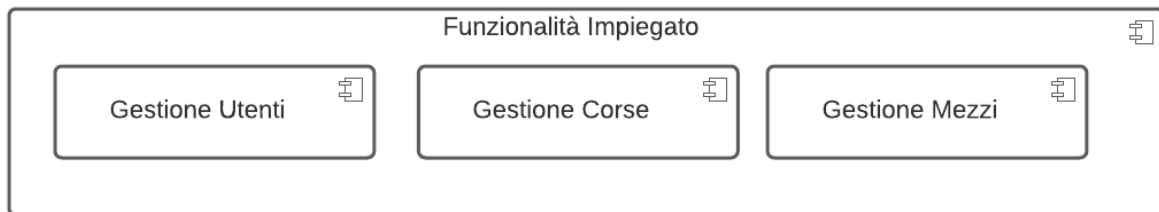
### 2.2. Decomposizione in sottoinsiemi

Tramite l'utilizzo dei component diagram di UML, si mostrano i sottosistemi di cui è formato il sistema.



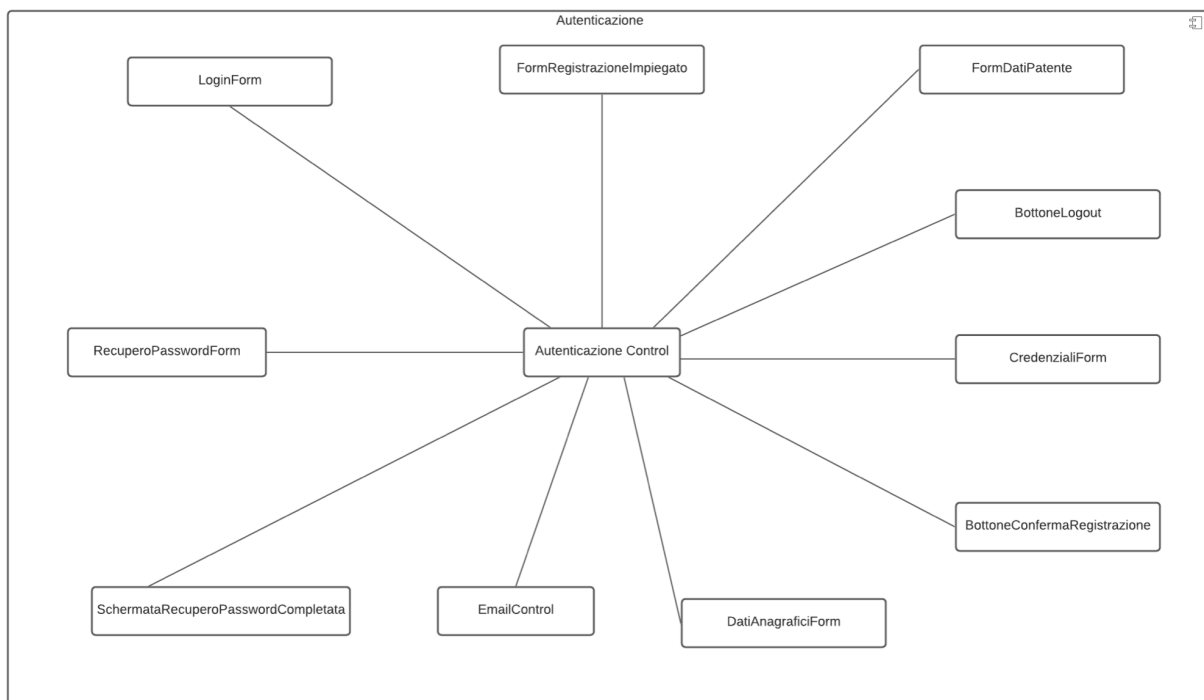
Per fornire una maggiore comprensione dei sottosistemi, di seguito verranno suddivisi in sottosistemi più piccoli.



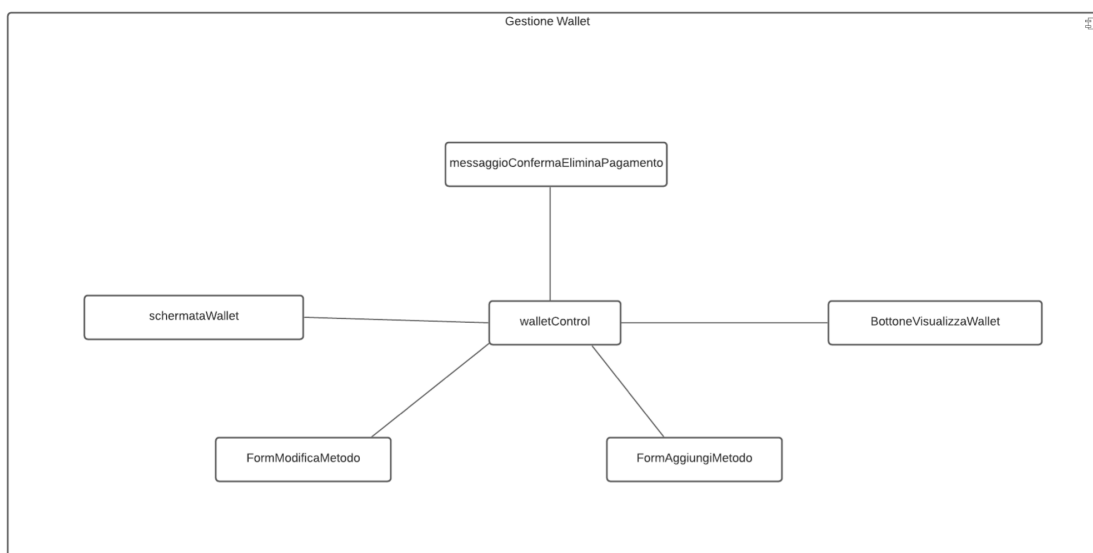
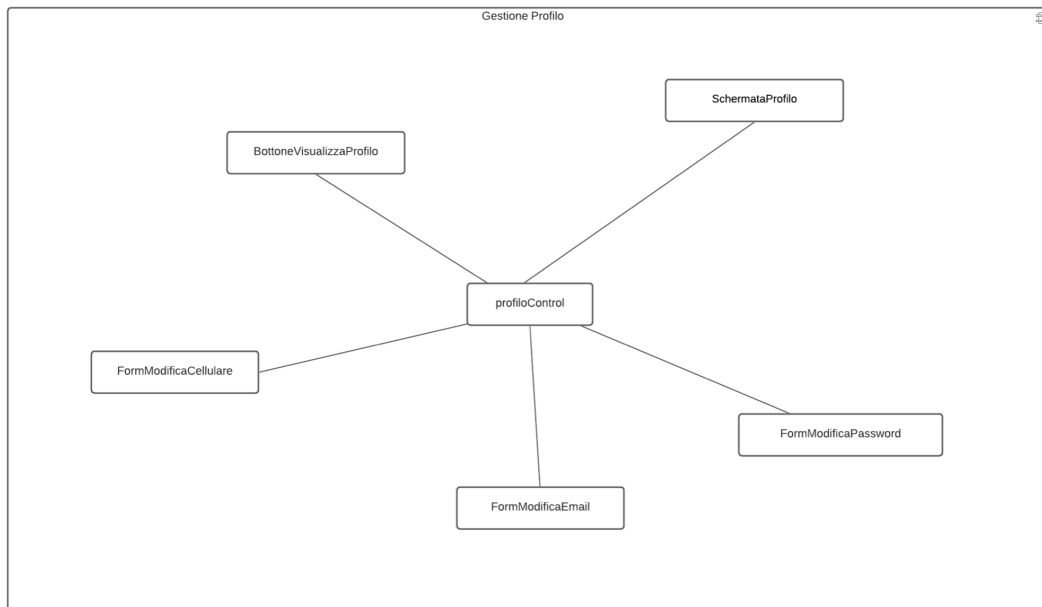


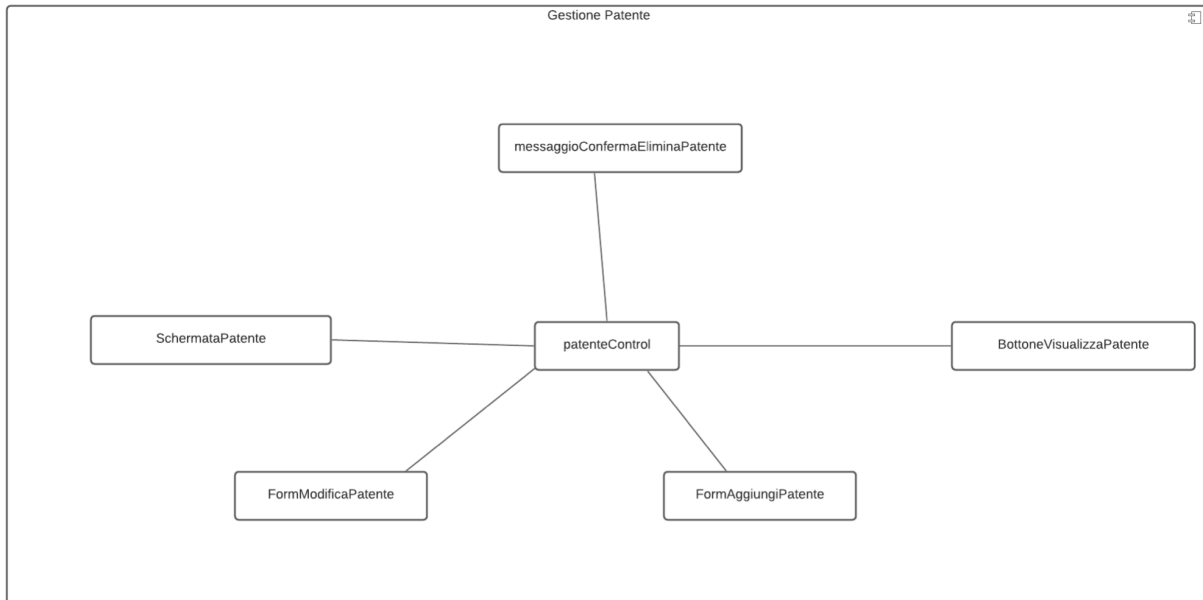
## 2.2.1. Mappatura degli oggetti nei sottoinsiemi

### Autenticazione

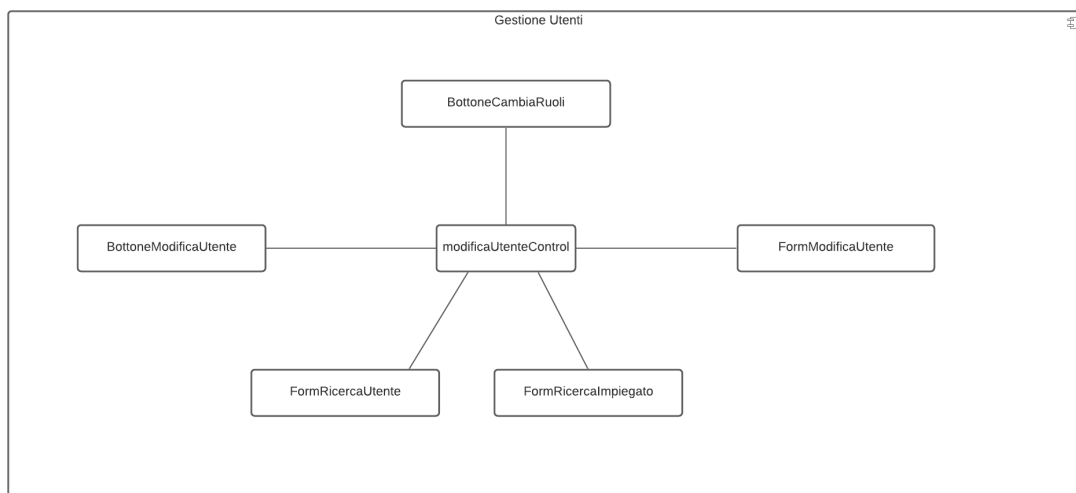


## Gestione Account

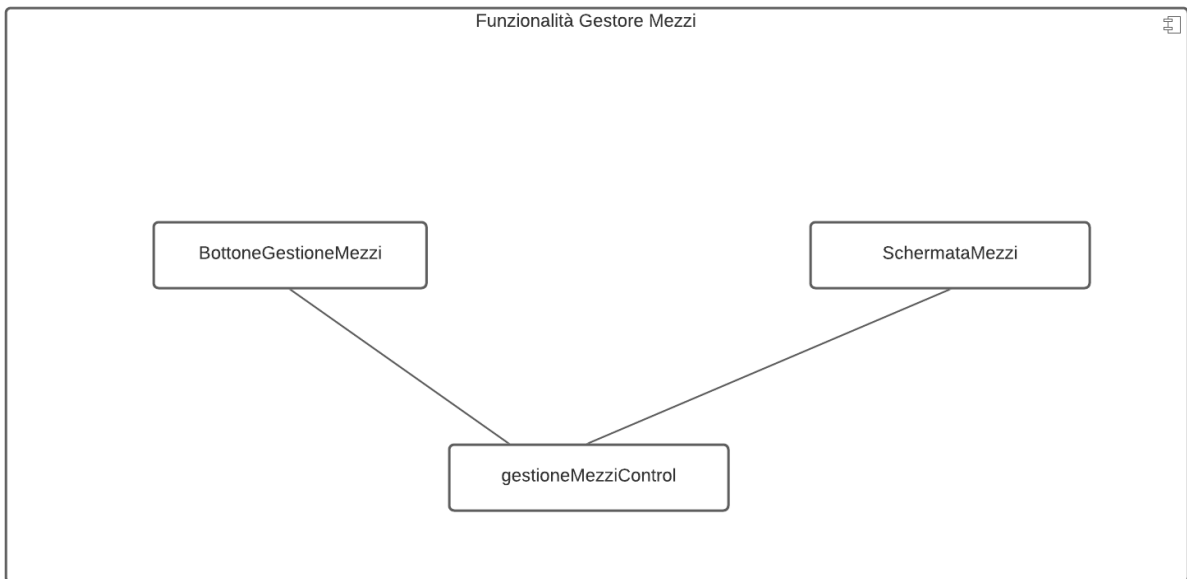
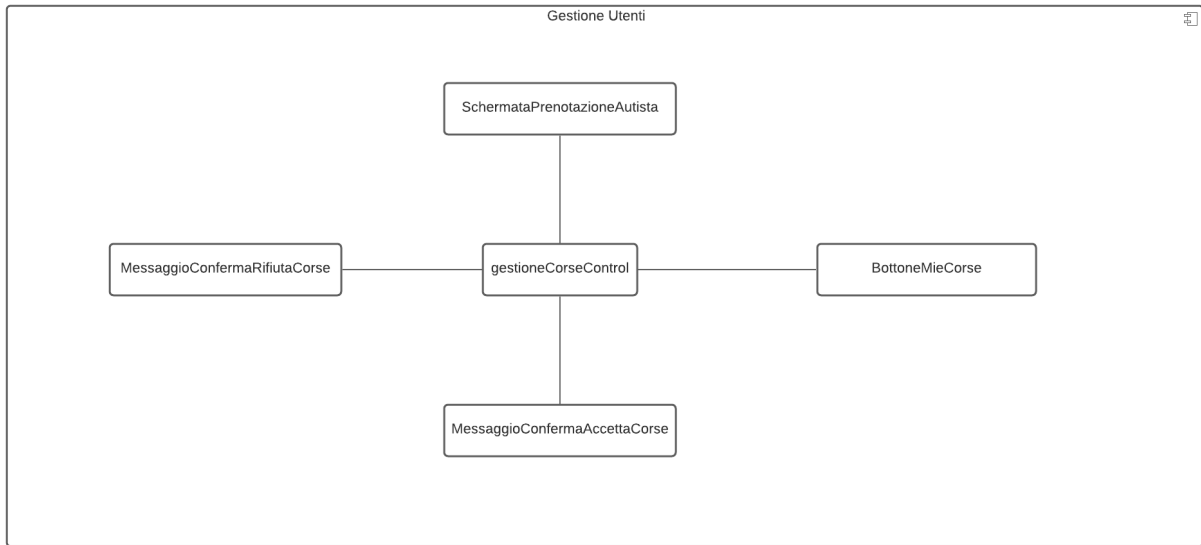




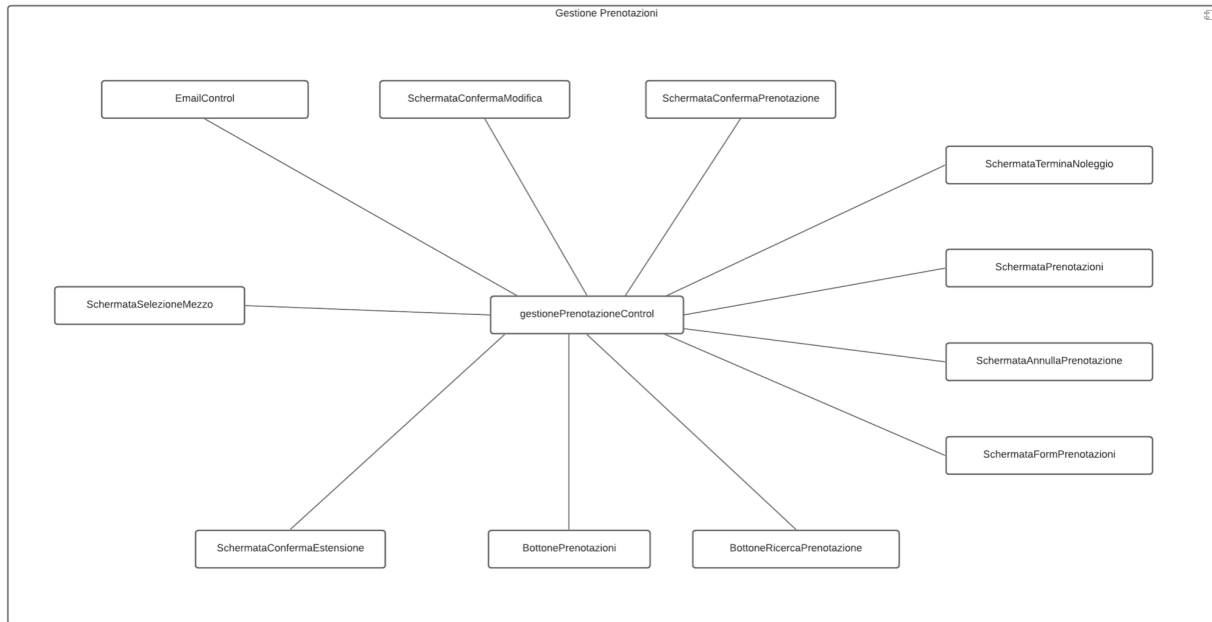
## Funzionalità impiegato





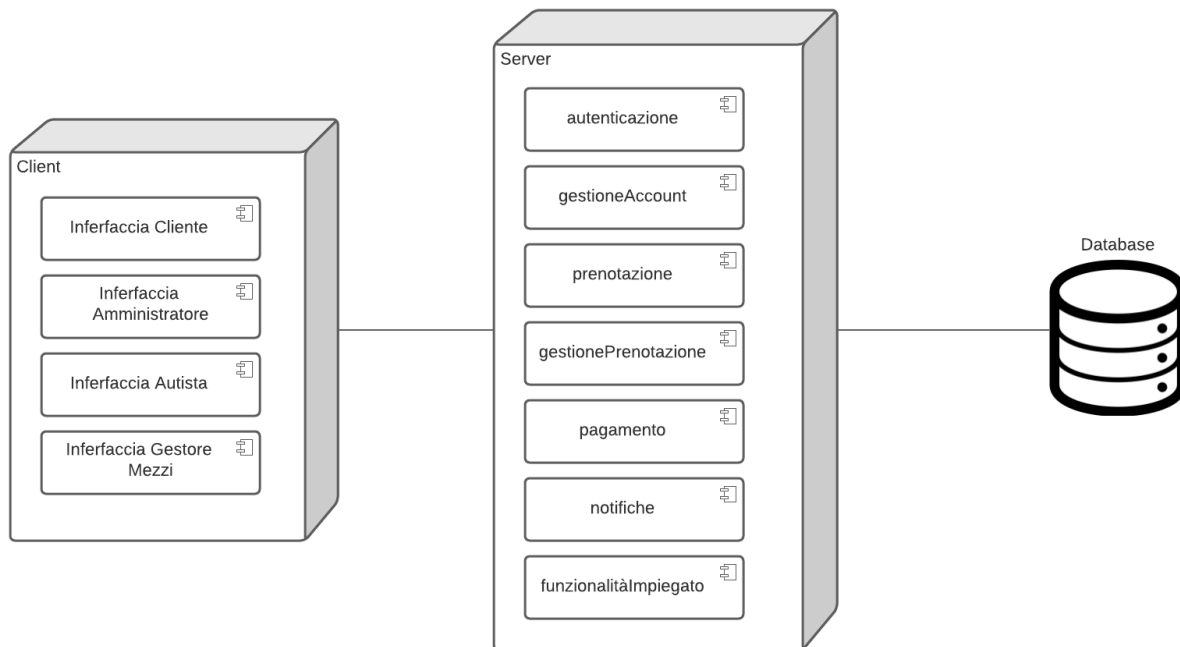


## Gestione prenotazione



### 2.3. Mappatura hardware/software

In modo concorde all'architettura client-server scelta, anche la struttura hardware segue lo stesso modello, essendo costituito da un server centrale e da più client che vengono identificati dai device degli utenti.



Al server centrale sono collegati il DBMS e tutti i client, che possono usufruire dei servizi a loro offerti.

In particolare, il nodo client è composto da diverse tipologie di sottosistemi, che suddividono e differenziano i ruoli che si possono avere all'interno del sistema.

## 2.4. Progetto del database

Poiché si è deciso di utilizzare un database non relazionale, in particolare MongoDB, basato su oggetti JSON, di seguito si riporta la struttura generale del database, suddiviso nelle collezioni che lo compongono.

### Deposito

#### **\_id : ObjectId**

Codice univoco del deposito, generato automaticamente da MongoDB.

**tipologia : String**

Rappresenta la tipologia del deposito, cioè parcheggio, dedicato ad auto e moto, o stallo, dedicato a bici e monopattini.

**nome : String**

Nome rappresentativo del deposito.

**posizione : Object**

Oggetto che contiene la posizione in coordinate del deposito.

**x : Double** Longitudine del deposito.

**y : Double** Latitudine del deposito.

**auto : Array[Object]**

Array che contiene un oggetto per ogni modello di auto dell'azienda.

**\_id : ObjectId** Codice univoco che identifica il modello di auto.

**targhe : Array[String]**

Array che contiene tutte le targhe delle auto di un particolare modello che si trovano nel parcheggio.

**moto : Array[Object]**

Array che contiene un oggetto per ogni modello di moto dell'azienda.

**\_id : ObjectId** Codice univoco che identifica il modello di moto.

**targhe : Array[String]**

Array che contiene tutte le targhe delle moto di un particolare modello che si trovano nel parcheggio.

**bici : Array[Object]**

Array che contiene un oggetto per ogni modello di bici dell'azienda.

**\_id : ObjectId** Codice univoco che identifica il modello di bici.

**targhe : Array[String]**

Array che contiene tutte le targhe delle bici di un particolare modello che si trovano nello stallo.

**monopattino : Array[Object]**

Array che contiene un oggetto per ogni modello di monopattino dell'azienda.

**\_id : ObjectId** Codice univoco che identifica il modello di monopattino.

**targhe : Array[String]**

Array che contiene tutte le targhe dei monopattini di un particolare modello che si trovano nello stallo.

**Veicolo****\_id : ObjectId**

Codice univoco del modello di veicolo, generato automaticamente da MongoDB.

**tipologia : String**

Rappresenta la tipologia del veicolo, cioè auto, moto, bici o monopattino

**marca : String**

Marca del veicolo.

**modello : String**

Nome del modello del veicolo.

**carburante : String**

Se specificato, rappresenta il tipo di carburante che utilizza il veicolo.

**cambio : String**

Se specificato, rappresenta il tipo di cambio del veicolo, cioè manuale o automatico.

**numeroPosti : Int**

Se specificato, rappresenta il numero di posti che il veicolo ha a disposizione.

**cilindrata : Int**

Se specificato, rappresenta il numero di cilindrata del motore.

**tariffaOraria : Int**

Rappresenta l'importo della tariffa oraria del veicolo.

**path : String**

Rappresenta il path dell'immagine rappresentativa del veicolo nella directory del server.

## Prenotazione

**\_id : ObjectId**

Codice univoco della prenotazione generato automaticamente da MongoDB.

**dataPrenotazione : Date**

Data di prenotazione.

**ritiro : Object**

Contiene le informazioni relative al ritiro del mezzo.

**idRitiro : ObjectId**

Reference al codice univoco di un deposito, nello specifico quello selezionato per il ritiro.

**nome : String** Nome del deposito di ritiro.

**data : Date** Data e ora del ritiro.

**consegna : Object**

Contiene le informazioni relative alla consegna del mezzo.

**idConsegna : ObjectId**

References al codice univoco di un deposito, nello specifico quello selezionato per la consegna.

**nome : String** Nome del deposito di consegna.

**data : Date** Data e ora della consegna.

**mezzo : Object**

**idMezzo : ObjectId** Reference al codice del mezzo prenotato.

**marca : String** Marca del mezzo prenotato.

**modello : String** Modello del mezzo prenotato.

**carburante : String** Se presente, tipo di carburante del mezzo prenotato.

**cilindrata : Int** Se presente, numero di cilindrata del mezzo prenotato.

**tariffa : Int** Costo orario del mezzo prenotato.

**tipologia : String**

Rappresenta la tipologia del veicolo prenotato, cioè auto, moto, bici o monopattino

**path : String** Percorso dell'immagine rappresentativa del veicolo nella directory del server.

**code : String** Targa del mezzo prenotato.

**tipologia : Object** Tipologia del mezzo prenotato.

**pagamento : Object**

Contiene le informazioni relative al pagamento della prenotazione.

**\_id : ObjectId** Reference al codice identificativo del metodo di pagamento utilizzato dal cliente.

**importoTotale : Int** Costo totale del servizio.

**autista : ObjectId** Reference al codice identificativo dell'autista, altrimenti, se non richiesto, viene impostato a false.

**stato : String** Rappresenta lo stato della prenotazione.

**idUtente : ObjectId** Reference al codice identificativo del cliente che ha effettuato la prenotazione.

## Utente

**\_id : ObjectId** Codice univoco dell'utente, generato automaticamente da MongoDB.

**nome : String** Nome dell'utente.

**cognome : String** Cognome dell'utente.

**dataNascita : String** Data di nascita dell'utente.

**Sesso : String** Sesso anagrafico dell'utente.

**luogoNascita : Object**

Oggetto che contiene i dati relativi al luogo di nascita dell'utente.

**nazione : String** Nazione di nascita dell'utente.

**regione : String**

Se specificata, rappresenta la regione di nascita dell'utente.

**provincia : String**

Se specificata, rappresenta la provincia di nascita dell'utente.

**citta : String**

Se specificata, rappresenta la città di nascita dell'utente.

**codiceFiscale : String** Codice fiscale dell'utente.



**credenziali : Object** Oggetto che contiene i dati relativi alle credenziali di accesso dell'utente.

**cellulare : String** Numero di cellulare dell'utente.

**email : String** Indirizzo email dell'utente.

**password : String** Password dell'utente criptata.

**accountStatus : Object**

Oggetto che rappresenta lo stato di attivazione dell'account dell'utente

**activatorKey : String**

Se specificata, chiave di attivazione dell'account dell'utente, altrimenti null.

**active : Boolean** Identifica lo stato dell'account cioè attivo o meno.

**metodiPagamento : Array[Object]** Array che contiene un oggetto per ogni metodo di pagamento associato all'account dell'utente.

**\_id : ObjectId**

Codice identificativo univoco del metodo di pagamento del cliente

**titolare : String** Titolare del metodo di pagamento.

**numeroCarta : String** Numero del metodo di pagamento.

**dataScadenzaCarta : String** Data di scadenza del metodo di pagamento..

**cvv : String** Card Verification Value del metodo di pagamento.

**patente : Object** Oggetto che contiene i dati relativi alla patente di guida associata all'account dell'utente, altrimenti null.

**numeroPatente : String** Numero della patente di guida.

**tipologiaPatente : String**

Tipologia della patente di guida, cioè AM, A, A1, A2 oppure B.

**dataScadenza : String** Data di scadenza della patente di guida.

**ufficioRilascio : String** Ufficio di rilascio della patente di guida.

## Notifiche

**\_id : ObjectId**

Codice univoco della notifica, generato automaticamente da MongoDB.

**title : String** Titolo della notifica.

**message : String** Messaggio della notifica.

**idUtente : ObjectId**

Reference al codice identificativo dell'utente a cui è diretta la notifica.

## 2.5. Accesso al sistema

L'accesso al sistema è consentito a qualsiasi utente, purché esso sia autenticato, previa registrazione.

Per capire se un utente è loggato o meno nel sistema vengono usati dei token di accesso e ogni tipo di utente (cliente, autista, gestore mezzi e amministrazione) ha a disposizione specifiche interfacce grafiche che gli permettono di accedere alle rispettive funzionalità.