



UNIVERSITÀ DEGLI STUDI DI PALERMO

---

# SYSTEM DESIGN DOCUMENT

Progetto "Noleggio auto/moto/bici/monopattino on-demand"



*Gli studenti:*

Salvatore Gabriele **Karra**  
Salvatore **Acquaviva**  
Maria **Rausa**  
Giuseppe **Maggio**

*Professori:*

Valeria Seidita  
Roberto Pirrone

---

Tesina 2020/21

# INDICE

1. **Architettura software attuale**
2. **Obiettivi di progettazione**
3. **Architettura software proposta**
  - 3.1. Panoramica
  - 3.2. Decomposizione in sottoinsiemi
  - 3.3. Mappatura hardware/software
  - 3.4. Progetto del database
4. **Accesso al sistema**

# 1. Architettura software attuale

Si suppone che non esista alcuna architettura con le stesse funzionalità attualmente adottate dall'azienda e che le attività di prenotazioni venivano svolte esclusivamente mediante comunicazione telefonica

## 2. Obiettivi di progettazione

- Il sistema deve essere correttamente visualizzato dalle ultime versioni stabili dei vari browser.
- Il sistema deve impedire l'inserimento di input non corretti da parte dell'utente (ad esempio la data di prenotazione precedente alla data attuale)
- Il sistema deve garantire funzionalità minimali di sicurezza, come la crittografia per la password di un Utente
- Il DBMS deve rimanere sempre attivo per consentire agli utenti una corretta fruizione dei servizi offerti

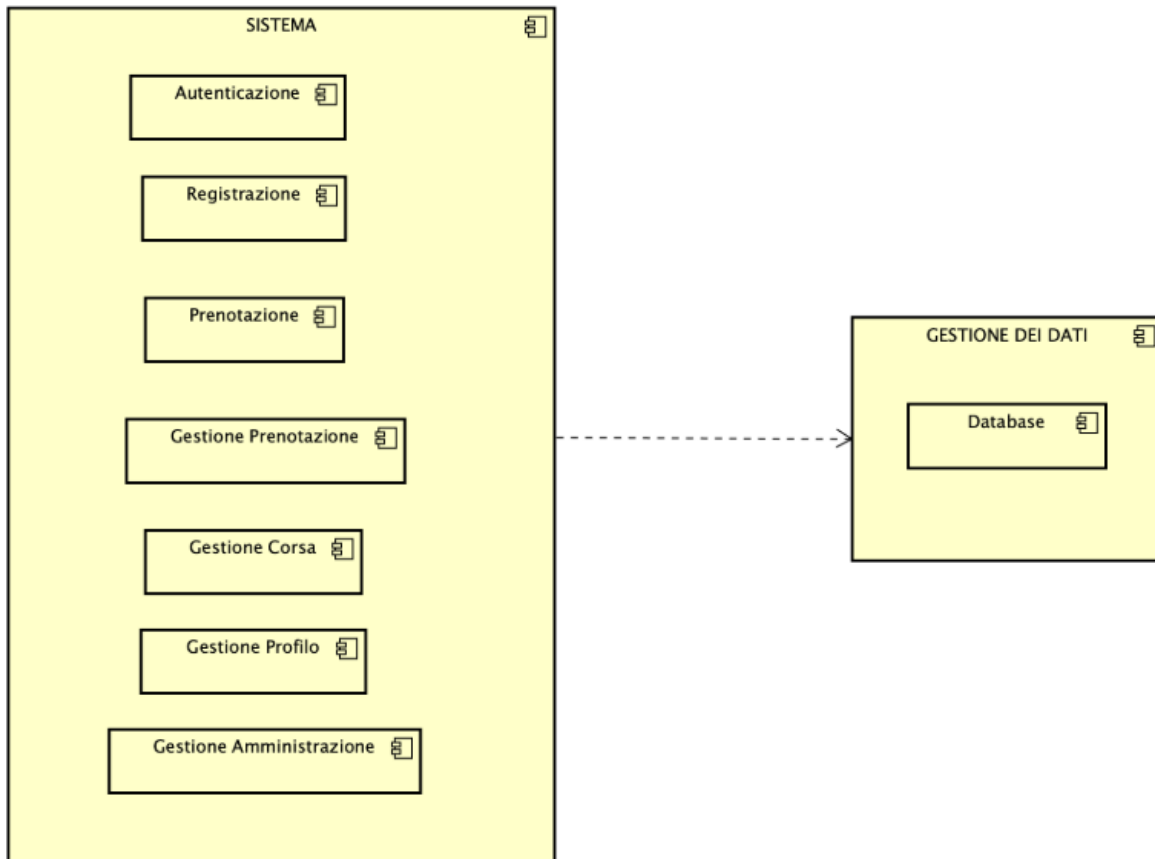
## 3. Architettura software proposta

### 3.1. Panoramica

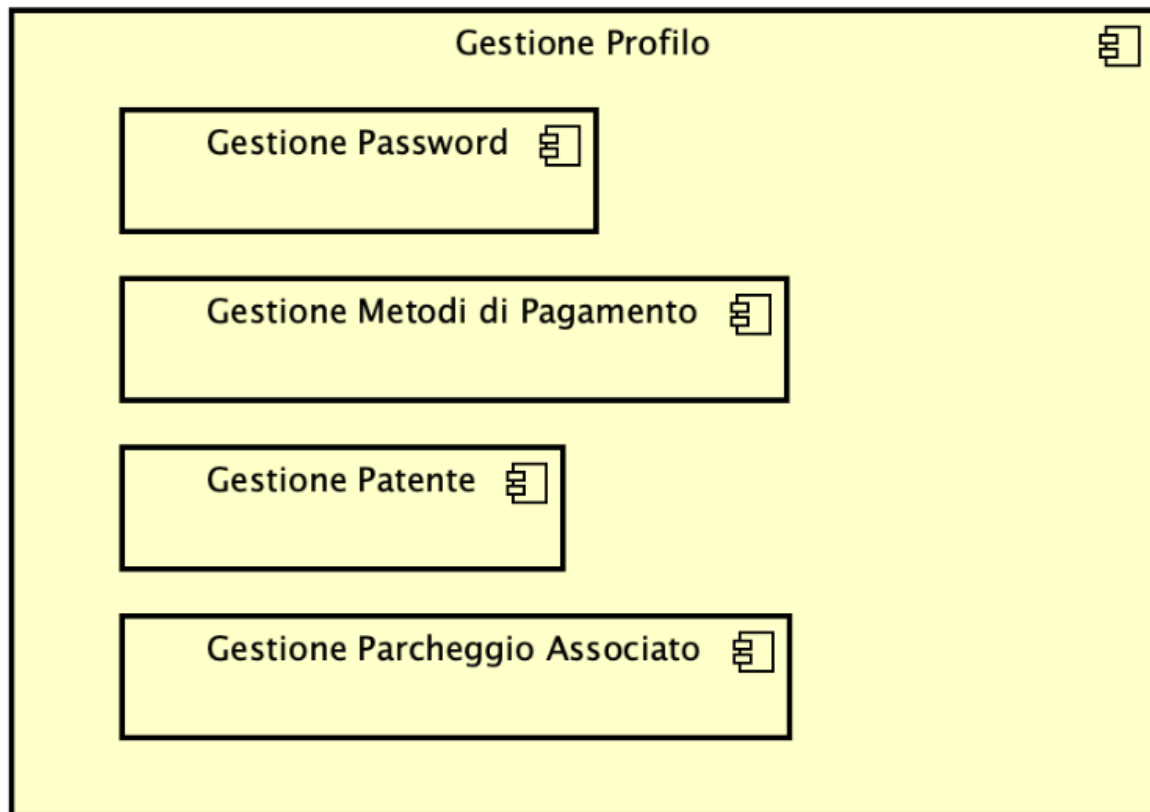
L'architettura software proposta è costituita da un unico nodo centrale che contiene tutti i dati di interesse e gli utenti interagiscono con il sistema tramite i loro dispositivi personali. Inoltre, gli utenti non hanno possibilità di interagire tra loro, quindi dovranno sempre fare richiesta al nodo 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.

## 3.2. Decomposizione in sottoinsiemi

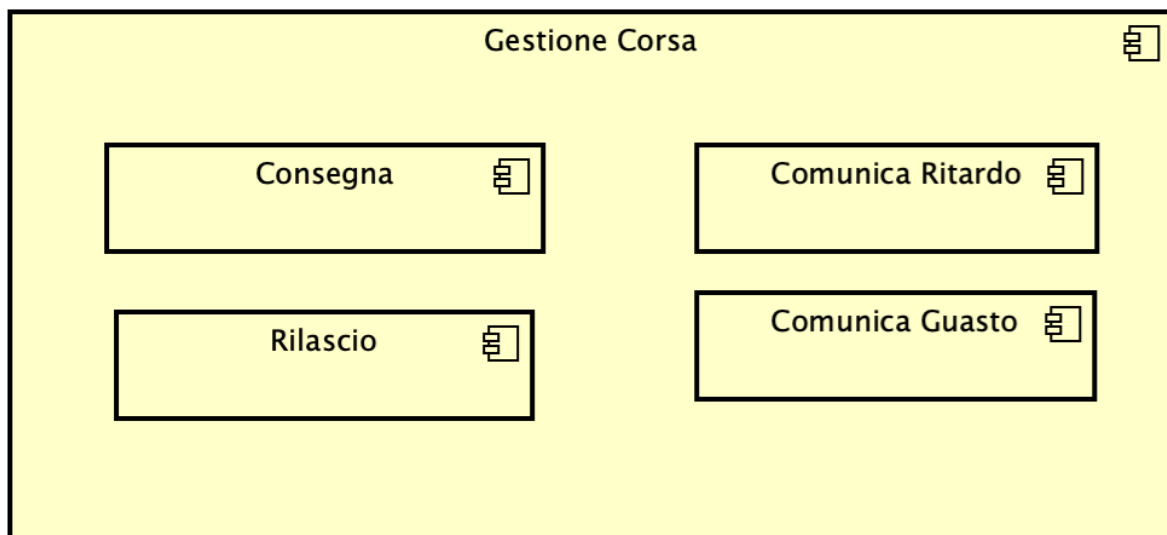
Tramite l'utilizzo dei component diagram di UML, si mostrano i sottoinsiemi di cui è formato il sistema.

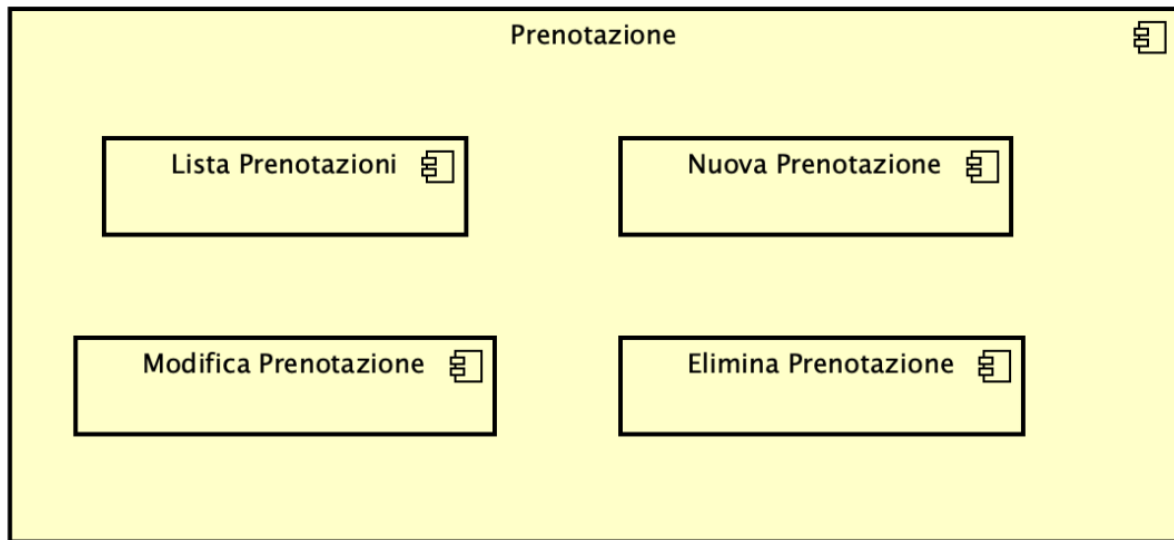


## System Design Document



\*Recupero Password





### 3.3. 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 usufruiscono del servizio collegandosi dal proprio device.

Al server centrale sono collegati il DBMS e tutti i client, che possono usufruire dei servizi offerti. In particolare il nodo client, è suddiviso in sottoinsiemi che suddividono e differenziano i ruoli che si possono avere all'interno del sistema.  
gestione amministrazione

### 3.4. Progetto del database

Poiché si è fatto uso di un database non relazionale (MongoDB), basato su oggetti JSON, si riporta di seguito la struttura generale del database, suddiviso negli Schema che lo compongono.

#### USERS:

**\_id: ObjectId**

Codice univoco dell'utente generato automaticamente da MongoDB

**Nome: String**

Rappresenta il nome dell'Utente

**Cognome: String**

Rappresenta il cognome dell'Utente

**data\_nascita: Date**

Data di nascita dell'Utente

## System Design Document

---

**Sesso: String**

Sesso dell'Utente

**luogo\_nascita: String**

Luogo di nascita dell'Utente

**provincia\_nascita: String**

Provincia di nascita dell'Utente

**codice\_fiscale: String**

Codice Fiscale dell'Utente

**email: String**

Email dell'Utente, che dovrà utilizzare per autenticarsi e ricevere eventuali codici di accesso o comunicazioni dall'azienda

**password: String**

Password scelta dall'utente in fase di registrazione per rendere il proprio account non accessibile agli altri

**numero\_patente: String**

Numero della patente del Cliente/Autista

**data\_rilascio\_patente: Date**

Data che fa riferimento a quando è stata rilasciata la patente al Cliente/Autista

**data\_scadenza\_patente: Date**

Data che fa riferimento alla scadenza della patente inserita da parte di Cliente/Autista

**ente\_rilascio: String**

L'ente che ha rilasciato la patente al Cliente/Autista

**Id\_parcheggio:**

Id del parcheggio associato all'Addetto al parcheggio

### VEICOLO

**\_id: ObjectId**

Codice univoco del Veicolo, generato automaticamente da MongoDB

**modello: String**

Modello del Veicolo

**marca: String**

Marca del Veicolo

**cilindrata: String**

Cilindrata del Veicolo

**n\_posti: String**

Numero di posti disponibili sul Veicolo

**n\_porte: String**

Numero di porte del veicolo

**targa: String**

Targa associata al Veicolo

**Id\_parcheggio: String**

Reference all'id del parcheggio nel quale si trova il veicolo

**descrizione: String**

Descrizione del Veicolo in merito ai propri benefit

**prezzo\_festivo: String**

## System Design Document

---

Prezzo associato al Veicolo durante i periodi festivi

**prezzo\_feriale: String**

Prezzo associato al Veicolo durante i periodi feriali

**stato: String**

Stato del Veicolo, può assumere: "Attivo", "Non attivo", "Bloccato", "Occupato"

### **PRENOTAZIONE**

**\_id: ObjectId**

Codice univoco della Prenotazione, generato automaticamente da MongoDB

**idCliente: String**

Reference all'id del Cliente che ha effettuato la prenotazione

**idVeicolo: String**

Reference all'id del Veicolo selezionato dal Cliente in fase di Prenotazione

**Partenza: String**

Luogo di partenza

**data\_partenza: Date**

Data di partenza

**ora\_partenza: String**

Ora di Partenza

**Destinazione: String**

Luogo di destinazione

**parcheggio\_destinazione: String**

Reference all'id del parcheggio di destinazione

**parcheggio\_partenza: String**

Reference all'ide del parcheggio di partenza

**data\_arrivo: Date**

Data di presunto arrivo a destinazione

**ora\_arrivo: String**

Ora presunta di arrivo a destinazione

**tipo\_veicolo: String**

Tipologia del Veicolo selezionato in fase di Prenotazione

**Stato: String**

Stato della Prenotazione, può assumere i valori "In-Completa", "Completata", "In Corso", "Terminata"

**idAutista: String**

Reference all'ide dell'Autista, qualora venga richiesto un'autista in fase di Prenotazione

### **PARCHEGGIO**

**\_id: ObjectId**

Codice univoco del Parcheggio, generato automaticamente da MongoDB

**nome: String**

Nome del parcheggio scelto in fase di inserimento nuovo Parcheggio, da parte dell'Amministratore

**indirizzo:String**

Indirizzo nel quale si trova il Parcheggio



## System Design Document

---

**numero\_civico: String**

Numero Civico dell'indirizzo del Parcheggio

**capienza\_auto: String**

Capienza massima del numero di autovetture nel Parcheggio

**capienza\_moto: String**

Capienza massima del numero di moto nel Parcheggio

**capienza\_bici: String**

Capienza massima del numero di biciclette nel Parcheggio

**capienza\_monopattini: String**

Capienza massima del numero di monopattini nel Parcheggio

### PAGAMENTO

**\_id: ObjectId**

Codice univoco del metodo di pagamento, generato automaticamente da MongoDB

**Id\_cliente: String**

Reference all'id del Cliente che ha inserito il metodo di pagamento

**numero\_carta: String**

Numero di carta del metodo di pagamento

**intestatario: String**

Intestatario del metodo di pagamento

**data\_scadenza: Date**

Data di scadenza del metodo di pagamento

**cvv: String**

Codice di sicurezza del metodo di pagamento

### NOTIFICHE

**\_id: ObjectId**

Codice univoco della notifica, generato automaticamente da MongoDB

**Id\_utente: String**

Reference all'id dell'Utente che l'ha generata

**tipo: String**

Tipologia della richiesta, può essere "Cliente" o "Autsista"

**titolo: String**

Titolo della notifica

**descrizione: String**

Testo della notifica

## 4. Accesso al sistema

L'accesso al sistema è consentito soltanto all'utente che si registra sulla piattaforma e si autentica mediante l'inserimento, nella Schermata di Accesso, della email e della password.

Un qualsiasi utente non registrato non può accedere ad alcun servizio. Nel caso in cui un utente abbia dimenticato la password per poter accedere al proprio account, può

## System Design Document

---

recuperarla, tramite il recupero password e l'inserimento del codice OTP che gli verrà inviato all'email registrata