

**Università degli Studi di Salerno**  
Corso di Ingegneria del Software

**AutoFlow**  
**System Design Document**  
**Versione 1.1**



Data: 08/11/2025

## Sommario

1. Introduzione.....	3
1.1. Scopo del sistema .....	3
1.2. Obiettivi e criteri di successo del progetto .....	4
1.3. Definizioni, acronimi e abbreviazioni.....	8
1.4. Riferimenti .....	9
1.5. Sintesi.....	9
2. Architettura corrente del software.....	10
3. Architettura software proposta.....	11
3.1. Panoramica .....	11
3.2 Decomposizioni sottosistemi.....	12
3.3 Mappatura Hardware/Software .....	14
3.4 Gestione dati persistenti.....	16
3.5 Controllo accessi e sicurezza .....	17
3.6 Controllo globale del software.....	20
3.7 Condizioni di boundary.....	21
4. Servizi dei sottosistemi .....	22
Glossario .....	26

## **1. Introduzione**

### **1.1. Scopo del sistema**

Il sistema AutoFlow è concepito per offrire una piattaforma web unificata dedicata alla gestione delle attività tipiche delle piccole e medie concessionarie automobilistiche. L'obiettivo principale è quello di mettere a disposizione delle concessionarie uno strumento moderno, integrato e tecnologicamente avanzato che consenta di superare l'attuale frammentazione dei processi, oggi spesso basati su fogli di calcolo, software scollegati tra loro, scambi informali di informazioni e archiviazione manuale della documentazione.

AutoFlow nasce per rispondere ai problemi organizzativi e operativi del settore, proponendo una soluzione che centralizzi in un unico ambiente tutte le fasi del ciclo di vendita del veicolo: dalla consultazione dello showroom alla configurazione autonoma dell'auto da parte del cliente, dalla gestione delle trattative alla generazione dei documenti amministrativi necessari per la conclusione dell'acquisto.

Il sistema permetterà infatti di semplificare e rendere più trasparente la gestione dello showroom fisico e virtuale, permettendo agli addetti di creare e aggiornare le schede dei veicoli, monitorare le disponibilità e seguire le trattative con maggiore efficienza. Allo stesso tempo, AutoFlow offrirà ai clienti un'esperienza digitale intuitiva e accessibile, consentendo loro di esplorare i veicoli disponibili, configurare il proprio modello ideale e inviare una proposta di acquisto direttamente dal portale, mantenendo sempre traccia delle interazioni con la concessionaria.

Un ulteriore elemento distintivo del sistema è la gestione centralizzata della documentazione: AutoFlow integrerà un meccanismo di generazione e archiviazione automatica dei documenti in formato PDF, come preventivi, proposte e fatture, permettendo di ridurre gli errori amministrativi e migliorare la conservazione delle informazioni nel tempo.

Il sistema sarà progettato per essere utilizzabile anche da utenti con competenze informatiche limitate, garantendo un'interfaccia semplice e una navigazione chiara sia per i clienti che per il personale della concessionaria.

## **1.2. Obiettivi e criteri di successo del progetto**

Il progetto AutoFlow ha come obiettivo principale la realizzazione di una piattaforma gestionale moderna e unificata, capace di ottimizzare il flusso operativo delle piccole e medie concessionarie automobilistiche. Il sistema non si limita a sostituire gli strumenti eterogenei oggi utilizzati, ma intende migliorare in modo significativo l'efficienza dei processi interni, offrendo una gestione più coerente delle trattative, dei veicoli e della documentazione. AutoFlow dovrà garantire un'interazione semplice tanto per i clienti, che possono configurare autonomamente un veicolo tramite lo showroom virtuale, quanto per gli addetti, che potranno monitorare e completare con precisione tutte le fasi della vendita.

La piattaforma dovrà inoltre assicurare affidabilità, sicurezza e integrità dei dati, implementando controlli automatici in grado di ridurre gli errori e semplificare le operazioni quotidiane. Obiettivo fondamentale sarà anche una gestione documentale strutturata, che consenta un'archiviazione ordinata e facilmente consultabile dei file generati durante il processo di vendita. Il successo del progetto si misurerà nella capacità di ridurre i tempi gestionali, migliorare la qualità dei dati, semplificare il lavoro degli addetti e rendere più trasparente l'esperienza del cliente, garantendo allo stesso tempo scalabilità e continuità operativa.

### **Portabilità**

AutoFlow deve poter funzionare senza problemi sui principali browser moderni e sui dispositivi comunemente utilizzati all'interno delle concessionarie e dagli utenti finali. L'obiettivo è garantire un accesso fluido allo showroom virtuale, alle funzioni di backoffice e ai moduli amministrativi, indipendentemente dal dispositivo utilizzato.

Assicurare la compatibilità con versioni aggiornate e stabili dei browser permette alla piattaforma di sfruttare funzionalità moderne e di offrire un'esperienza coerente a venditori, amministratori e clienti.

### **Robustezza**

La piattaforma deve proteggere con cura sia i dati sensibili dei clienti sia le comunicazioni tra browser e server. L'adozione del protocollo TLS 1.3 garantisce connessioni sicure mentre gli algoritmi di hashing avanzati proteggono credenziali e informazioni personali archiviate nel database.

AutoFlow integra inoltre meccanismi di prevenzione contro le vulnerabilità più comuni — come SQL Injection, XSS e tentativi di accesso non autorizzato — assicurando un ambiente sicuro anche durante operazioni critiche come la generazione delle fatture e la gestione delle proposte.

### **User-Friendliness**

La piattaforma è progettata per essere accessibile e semplice da utilizzare, anche per utenti con esigenze particolari o competenze digitali limitate.

Test periodici con strumenti come Lighthouse e tecnologie assistive (ad esempio JAWS) assicurano che l'esperienza utente rispetti gli standard di accessibilità, facilitando la navigazione, l'inserimento dei dati e la consultazione dei documenti.

AutoFlow mira a offrire un'interfaccia chiara e intuitiva, riducendo i tempi di apprendimento e aumentando la soddisfazione degli utenti.

## **High Performance**

AutoFlow deve garantire caricamenti rapidi, pagine reattive e operazioni fluide, anche in presenza di un elevato numero di veicoli, proposte e dati amministrativi.

Tecniche come caching, minimizzazione del codice, utilizzo di CDN e caricamento differito delle risorse ottimizzano la velocità complessiva del sistema. Ciò permette a venditori e clienti di svolgere le proprie attività senza attese inutili, soprattutto durante momenti di maggiore affluenza, come weekend o periodi di promozione.

## **Adaptability**

Il sistema deve mantenere piena usabilità su schermi con risoluzioni differenti, dai monitor dei computer in concessionaria fino ai laptop dei clienti che consultano lo showroom virtuale da casa.

L'interfaccia di AutoFlow è progettata per adattarsi automaticamente allo spazio disponibile, garantendo un'esperienza coerente sia durante la configurazione del veicolo sia nelle procedure amministrative.

Test su vari formati e dispositivi aiutano a prevenire problemi di visualizzazione e interazione.

## **Affidabilità**

AutoFlow deve garantire continuità operativa anche con carichi elevati, supportando almeno 1500 utenti contemporanei senza degradare prestazioni o stabilità.

L'obiettivo è assicurare che venditori e clienti possano utilizzare il sistema senza interruzioni, in particolare durante le ore di apertura della concessionaria.

## **Fault Tolerance**

La piattaforma integra strategie avanzate per prevenire errori e attacchi informatici, mantenendo l'operatività anche in condizioni critiche.

Protezione da vulnerabilità:

- prevenzione SQL Injection
- prevenzione XSS
- controlli server-side su tutte le operazioni sensibili

Gestione dei guasti:

- log dettagliati per analizzare eventuali anomalie
- backup regolari per garantire il recupero dei dati in caso di problemi

Questi elementi permettono allo staff di identificare e risolvere rapidamente eventuali criticità.

## **Interfacce ben definite**

L'accesso alle funzioni della piattaforma deve essere controllato rigorosamente da un sistema di permessi basato sui ruoli (cliente, addetto vendite, amministratore).

Ogni utente può visualizzare e modificare solo le sezioni di propria competenza, garantendo sia la sicurezza dei dati sia una maggiore chiarezza operativa.

In AutoFlow, questo significa ad esempio che un cliente può consultare le proprie configurazioni, ma non accedere ai moduli interni dello showroom o della fatturazione.

### **Ease of Use**

L'interfaccia di AutoFlow trae ispirazione da applicativi familiari al pubblico target, rendendo l'esperienza più naturale e immediata.

Sessioni di usability testing con vendori e clienti reali permettono di rilevare eventuali criticità nel flusso operativo, ottimizzando menu, form e procedure.

Il risultato è una piattaforma che riduce errori, velocizza la formazione del personale e accompagna l'utente in tutte le fasi della trattativa.

### **Efficienza**

AutoFlow riduce il carico sul database tramite strategie di caching, ottimizzazione delle query e memorizzazione intelligente dei contenuti.

Queste tecniche permettono tempi di risposta più rapidi e riducono il carico durante operazioni frequenti come:

- caricamento dello showroom
- aggiornamento delle schede veicolo
- consultazione delle statistiche
- generazione delle proposte

### **Scalabilità**

La piattaforma è pensata per crescere nel tempo, sia verticalmente sia orizzontalmente.

Poiché un sistema gestionale per concessionarie può espandersi includendo nuove sedi, più utenti o maggiori volumi di dati, AutoFlow è progettato per integrare in futuro un'autenticazione completamente stateless basata su JWT o OAuth.

Questo approccio riduce la necessità di sincronizzare lo stato tra più server, facilitando la distribuzione del sistema in ambienti multi-nodo.

### **Rapid Development**

Per velocizzare il rilascio della prima versione, AutoFlow adotta inizialmente un sistema di autenticazione tradizionale basato su form e cookie, minimizzando i dati conservati in sessione.

Ciò consente uno sviluppo più rapido, senza rinunciare alla possibilità di evolvere in futuro verso tecnologie più scalabili come JWT.

Questa strategia permette di consegnare rapidamente una piattaforma completa nella sua funzionalità base, per poi perfezionarla in iterazioni successive.

## **Cost-effectiveness**

AutoFlow è progettato per offrire il massimo risultato con il minimo investimento necessario.

L'architettura del sistema bilancia qualità, prestazioni e costi, privilegiando scelte tecnologiche che riducono le spese operative e semplificano manutenzione e aggiornamenti.

L'obiettivo è permettere anche alle concessionarie di piccole dimensioni di adottare una soluzione moderna e professionale, senza costi proibitivi di infrastruttura.

## **Trade-off**

Nel processo di progettazione del sistema è stato necessario valutare con attenzione diversi trade-off, ciascuno legato alle priorità e ai limiti imposti dal contesto applicativo. Queste scelte hanno coinvolto aspetti cruciali come l'allocazione delle risorse, il livello di complessità tecnica ritenuto accettabile e l'equilibrio tra costi e benefici. L'obiettivo è stato quello di individuare soluzioni che garantissero buone prestazioni e funzionalità adeguate, senza incidere negativamente sulla qualità complessiva della piattaforma né comprometterne la sostenibilità nel lungo periodo.

## **Affidabilità vs Cost-effectiveness**

Per garantire un sistema stabile, reattivo e in grado di gestire picchi di utilizzo, sarebbe ideale adottare infrastrutture cloud scalabili e sistemi avanzati di monitoraggio.

Tuttavia, per contenere i costi si può partire da configurazioni più semplici — come hosting condivisi o macchine locali — consapevoli che ciò può ridurre la resilienza del sistema sotto carico elevato.

AutoFlow adotta un equilibrio che permette una buona stabilità senza generare spese eccessive, lasciando aperta la possibilità di scalare in futuro.

## **Scalabilità vs Rapid Development**

Implementare sin da subito un sistema di autenticazione stateless basato su token (JWT/OAuth) favorirebbe la scalabilità orizzontale ma aumenterebbe i tempi e la complessità dello sviluppo.

AutoFlow sceglie invece un approccio più snello per le prime versioni, basato su sessioni e cookie, così da accelerare il rilascio.

In seguito, quando la piattaforma crescerà o verrà distribuita su più sedi, sarà possibile migrare verso un sistema completamente stateless senza impatti significativi.

### **1.3. Definizioni, acronimi e abbreviazioni**

Di seguito è fornito un elenco degli acronimi, abbreviazioni con le relative definizioni utilizzati in questo documento:

Termine	Definizione
AES-256	Advanced Encryption Standard a 256 bit
AWS	Amazon Web Service
DDoS	Distributed Denial of Service
DNS	Domain Name System
FR	Requisito funzionale
HTTP	Hypertext Transfer Protocol
JDBC	Java Database Connectivity
JSF	JavaServer Faces
JWT	JSON Web Token
NA	Not Accessible
OAuth	Open Authorization
PHP	Hypertext Preprocessor
SSL	Secure Sockets Layer
TLS	Transport Layer Security
XSS	Cross-Site-Scripting

## **1.4. Riferimenti**

Il presente progetto si basa sull'analisi e il confronto con piattaforme di gestione della didattica già consolidate e affermate nel settore, le quali hanno dimostrato notevole efficacia. Tra queste, un punto di riferimento significativo è rappresentato dalle soluzioni sviluppate da Argo per la gestione della didattica.

Di seguito si presenta un elenco dei documenti chiave del progetto a cui si fa esplicito riferimento:

- Problem Statement: documento che definisce i problemi principali che il progetto intende affrontare e risolvere.
- Requirements Analysis Document (RAD): documento che definisce e analizza i requisiti funzionali e non funzionali di un sistema, descrivendo le necessità degli utenti e i vincoli progettuali.

Oltre ai documenti del progetto, si fa riferimento a opere di letteratura tecnica e risorse online che hanno contribuito allo sviluppo metodologico e concettuale di questo lavoro:

- Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition di Bernd Bruegge & Allen H. Dutoit.

## **1.5. Sintesi**

In questo paragrafo sono stati affrontati temi legati allo sviluppo di una piattaforma gestionale per concessionarie automobilistiche, con particolare attenzione alla sostenibilità operativa, alla semplificazione dei processi interni e al miglioramento dell'esperienza utente per clienti e personale di vendita.

Viene discusso il bilanciamento tra affidabilità del sistema e contenimento dei costi infrastrutturali, evidenziando le sfide tecniche e le opportunità di evoluzione futura. Si analizzano inoltre le strategie per garantire scalabilità, sicurezza e continuità del servizio, considerando la necessità di un percorso di crescita graduale del progetto.

Infine, si sottolinea l'importanza di progettare una soluzione flessibile, capace di adattarsi a concessionarie con esigenze differenti e pronta a evolversi nel tempo per integrare nuove funzionalità e miglioramenti.

La restante parte del documento presenta una descrizione dettagliata dei sottosistemi individuati, dell'architettura della piattaforma AutoFlow e delle modalità con cui vengono gestite le principali componenti operative.

## 2. Architettura corrente del software

Il sistema **AutoFlow** è attualmente in fase di sviluppo e, allo stato corrente, non esiste ancora alcuna implementazione funzionante. La piattaforma rappresenta una soluzione totalmente nuova, progettata per digitalizzare e semplificare i processi delle piccole e medie concessionarie automobilistiche. Non essendoci un sistema di riferimento, la progettazione si basa sull'analisi delle esigenze operative delle concessionarie e sulle funzionalità individuate nei documenti progettuali, con l'obiettivo di offrire una piattaforma moderna, scalabile e facilmente estendibile nel tempo.

AutoFlow è pensato per adottare un'architettura distribuita che separi chiaramente la logica applicativa dal livello di presentazione, seguendo le buone pratiche suggerite nei documenti progettuali del sistema. Il backend sarà implementato in Java con Spring Boot, mentre il frontend utilizzerà tecnologie JavaScript/TypeScript come React o Angular, garantendo così un'interfaccia responsiva e facilmente adattabile.

Il database sarà relazionale (PostgreSQL o MySQL), centralizzato e strutturato per garantire integrità e tracciabilità delle operazioni.

Coerentemente con i requisiti non funzionali, l'applicazione farà uso di comunicazioni sicure tramite TLS 1.3, salvataggio dei dati sensibili mediante hashing (es. SHA-512) e adozione di meccanismi di controllo degli accessi basati sui ruoli (cliente, addetto vendite, amministratore). La piattaforma sarà progettata per mantenere una buona scalabilità, in modo da poter supportare l'aumento progressivo di veicoli, utenti e trattative.

Un aspetto centrale del sistema riguarda la gestione e la tracciabilità dei dati: veicoli, proposte, configurazioni e documenti (preventivi e fatture in PDF) verranno archiviati in modo strutturato all'interno di directory organizzate, secondo quanto definito nei requisiti e negli scenari. L'obiettivo è garantire affidabilità, reperibilità immediata della documentazione e storicizzazione completa delle attività, pur mantenendo basse le complessità infrastrutturali nella prima versione.

In assenza di una piattaforma di riferimento, AutoFlow propone dunque una soluzione moderna e modulare, capace di evolvere nel tempo e adattarsi alle necessità delle concessionarie, mantenendo un equilibrio tra costi, prestazioni e facilità di sviluppo.

### **3. Architettura software proposta**

#### **3.1. Panoramica**

Il sistema AutoFlow sarà progettato come una piattaforma gestionale moderna e unificata, pensata per digitalizzare l'intero ciclo operativo delle piccole e medie concessionarie automobilistiche. L'architettura software adottata segue un modello three-tier, articolato in tre strati principali: interfaccia utente, logica applicativa e gestione dei dati. Questa suddivisione garantisce una chiara separazione delle responsabilità, favorendo scalabilità, manutenibilità ed estensibilità del sistema nel tempo.

La natura distribuita dell'architettura consente di gestire in modo indipendente ciascun layer, permettendo ai diversi sottosistemi: showroom virtuale, gestione veicoli, trattative, configurazioni, documentazione e fatturazione, di operare in sinergia per assicurare un'esperienza d'uso fluida e coerente, sia per i clienti sia per il personale interno. Ogni sottosistema è progettato per coprire specifiche funzionalità operative, come la gestione delle credenziali e dell'autenticazione, la registrazione e configurazione dei veicoli, la creazione e aggiornamento delle proposte di vendita, la generazione dei documenti PDF, l'aggiornamento dello stato dei veicoli e l'analisi delle statistiche di vendita.

Il sottosistema dedicato all'interfaccia utente è concepito per essere intuitivo e pienamente accessibile anche a utenti con competenze informatiche limitate, come clienti e addetti vendite. La gestione dell'autenticazione e della sicurezza è garantita da controlli lato server, algoritmi di hashing robusti e protocolli di comunicazione cifrati, assicurando la protezione dei dati personali e la conformità alle normative vigenti.

La piattaforma integra, inoltre, strumenti che supportano tutte le operazioni quotidiane di concessionaria: dalla configurazione autonoma del veicolo da parte del cliente, alla gestione delle trattative in sede, fino all'emissione delle fatture e alla loro archiviazione automatica. Il database relazionale svolge un ruolo centrale, archiviando in modo sicuro e strutturato informazioni relative a veicoli, clienti, configurazioni, proposte e documenti amministrativi. L'adozione di un'architettura modulare e scalabile consente al sistema di mantenere prestazioni elevate anche in presenza di numerosi utenti simultanei e grandi volumi di dati, garantendo al tempo stesso affidabilità, continuità del servizio e possibilità di estensioni future.

### **3.2 Decomposizioni sottosistemi**

Sono stati individuati e definiti i seguenti sottosistemi, che rappresentano le principali componenti funzionali e strutturali, all'interno delle quali verrà decomposto il sistema AutoFlow, al fine di facilitarne lo sviluppo, la gestione e la manutenzione:

#### **User Interface**

Il sottosistema User Interface gestisce tutte le componenti visive della piattaforma, come showroom virtuale, dashboard e moduli di inserimento dati. Fornisce agli utenti un'interazione chiara e intuitiva, garantendo coerenza grafica e adattamento ai diversi dispositivi.

#### **Authentication**

Il sottosistema Authentication si occupa della verifica dell'identità degli utenti attraverso login, attivazione via e-mail e recupero password. Applica controlli di sicurezza come l'hashing delle credenziali e la validazione delle password, assicurando accessi protetti.

#### **User Management**

Il sottosistema User Management gestisce la creazione, la modifica e la disattivazione degli account per clienti, addetti vendite e amministratori. Assicura anche la corretta assegnazione dei ruoli e dei permessi, permettendo un controllo accurato dei livelli di accesso.

#### **Showroom & Vehicle Management**

Questo sottosistema permette l'inserimento, la modifica e la pubblicazione delle schede veicolo, verificando eventuali duplicati e gestendo lo stato (disponibile, non disponibile, venduto). Mantiene allineati lo showroom online e l'inventario reale della concessionaria.

#### **Configuration**

Il sottosistema Configuration gestisce il processo guidato di configurazione del veicolo, consentendo al cliente o allo staff di scegliere modello, optional e caratteristiche. Calcola dinamicamente il prezzo finale e collega la configurazione alla proposta commerciale.

#### **Proposals & Sales**

Il sottosistema Proposals & Sales gestisce l'intero flusso delle proposte, dalla creazione alla presa in carico, fino alla conferma o all'annullamento. Ogni proposta è tracciata tramite codice univoco e segue un percorso controllato fino alla finalizzazione della vendita.

#### **Document Management & PDF Generation**

Questo sottosistema genera automaticamente documenti PDF come preventivi e fatture, applicando numerazione progressiva. Archivia i documenti in strutture ordinate per anno e mese, rendendoli consultabili da staff e clienti in modo semplice e tracciabile.

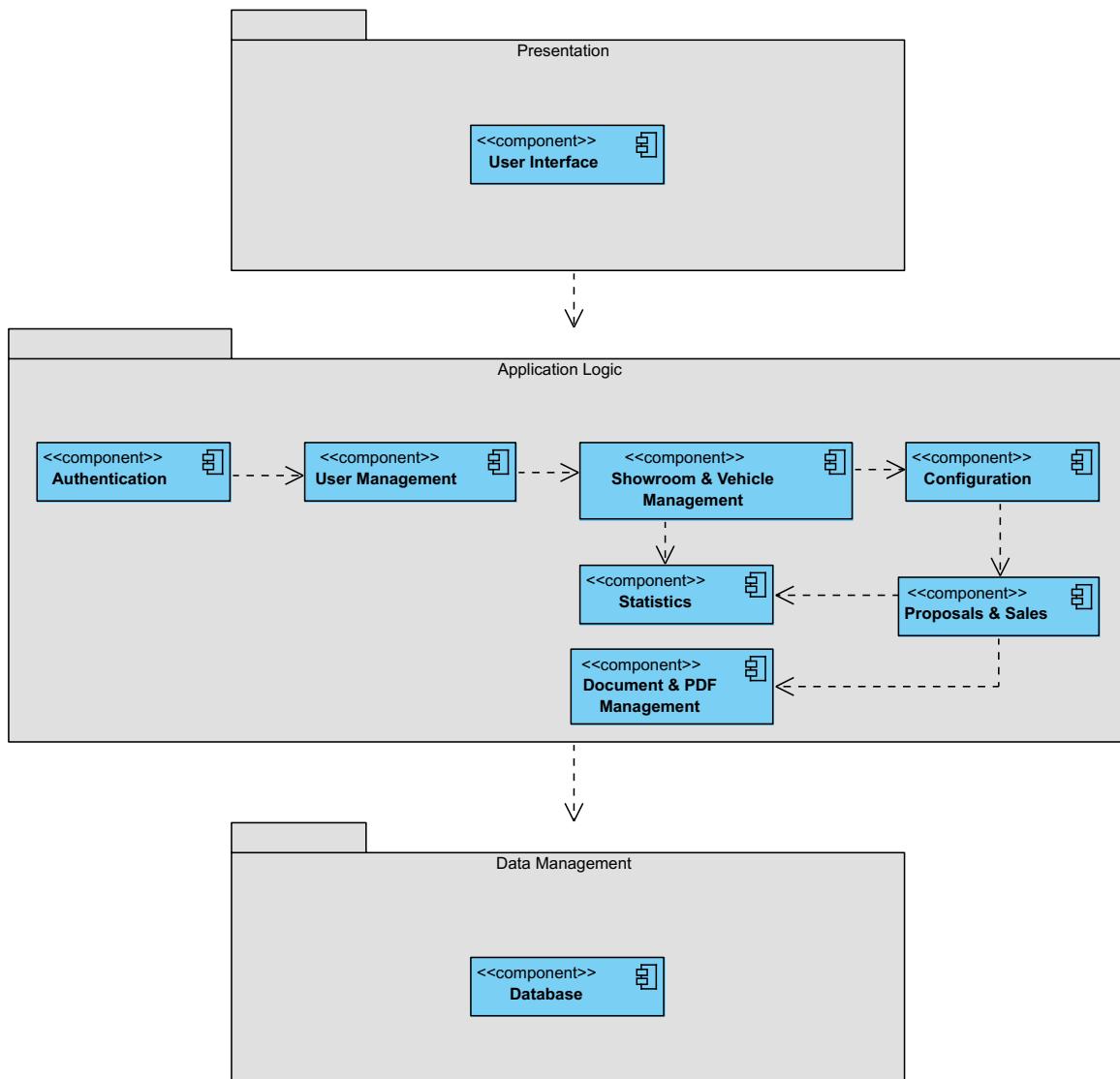
## Statistics

Il sottosistema Statistics elabora e visualizza i principali indicatori relativi a vendite, trattative e disponibilità dei veicoli. Fornisce grafici e tabelle dinamiche utili all'analisi delle performance della concessionaria.

## Database

Il sottosistema Database gestisce la memorizzazione persistente di tutte le informazioni relative a utenti, veicoli, configurazioni, proposte e documenti. Garantisce integrità referenziale, sicurezza dei dati e supporto alle operazioni degli altri sottosistemi.

Di seguito il diagramma dei componenti suddiviso nei layer Presentation, Application Logic, Data Management, per una migliore leggibilità sono omesse le dipendenze tra la user interface e i componenti della application logic. Sono anche omesse le dipendenze dell'application logic con il database.



### **3.3 Mappatura Hardware/Software**

Per AutoFlow si è scelto di adottare un'architettura three-tier distribuita, che garantisce scalabilità, manutenibilità e una chiara separazione tra interfaccia utente, logica di business e livello dati. L'architettura segue un modello chiuso: ogni livello può comunicare esclusivamente con quello immediatamente sottostante, evitando dipendenze indesiderate e incrementando la sicurezza complessiva della piattaforma.

La distribuzione software avverrà tramite pacchetti JAR eseguibili con web container integrato (Spring Boot + Tomcat embedded), in linea con quanto previsto dai requisiti di deployment. L'infrastruttura potrà essere ospitata su server Linux o, in configurazioni più avanzate, su macchine virtuali fornite da provider cloud (ad esempio Microsoft Azure), permettendo una rapida scalabilità sia verticale sia orizzontale.

Il backend AutoFlow, implementato in Java, dialogherà con il DBMS relazionale (PostgreSQL/MySQL) tramite JDBC, gestendo operazioni critiche come autenticazione, gestione veicoli, trattative, configurazioni, proposte e generazione delle fatture PDF. Per garantire isolamento e resilienza, il database sarà preferibilmente collocato su un nodo dedicato, così da evitare che eventuali malfunzionamenti del livello applicativo possano compromettere la consistenza dei dati persistenti.

Sebbene AutoFlow non preveda, nella sua prima versione, l'utilizzo di CDN o infrastrutture avanzate, l'architettura è predisposta per integrare in futuro reti di distribuzione dei contenuti e meccanismi di caching distribuito, così da ottimizzare il caricamento delle risorse statiche (immagini dei veicoli, PDF generati, contenuti dello showroom).

Le richieste dei client avverranno tramite HTTPS cifrato (TLS 1.3), indirizzate al web server dell'applicazione. Il backend eseguirà le operazioni richieste e, quando necessario, interrogherà il database tramite JDBC o accederà al file system per la gestione dei PDF generati. L'archiviazione automatica di preventivi e fatture (organizzate per anno e mese) segue quanto definito negli scenari del sistema.

Come tutte le architetture anche la three-tier presenta vantaggi e svantaggi:

#### **Vantaggi**

##### **Modularità e Manutenibilità**

- Ogni livello ha responsabilità chiare (UI, logica applicativa, database), rendendo più semplice intervenire su singoli moduli.
- Gli aggiornamenti lato server non impattano il frontend e viceversa.

##### **Scalabilità**

- Possibilità di distribuire più istanze del backend in parallelo, bilanciando il carico in caso di numerosi utenti simultanei.
- Il database, isolato su nodo dedicato, può essere scalato indipendentemente dagli altri livelli.

##### **Affidabilità**

- La separazione dei livelli riduce il rischio di guasti globali.
- È possibile introdurre backup, replica dati e meccanismi di failover per garantire continuità operativa.

## Sicurezza

- Tutti gli accessi ai dati avvengono tramite la logica applicativa, che applica controlli, hashing (SHA-512) e policy di validazione.
- Il database non è accessibile direttamente dall'esterno, ma solo tramite il backend.

## Flessibilità

- Ogni livello può adottare tecnologie differenti:
  - Frontend: React/Angular
  - Backend: Java + Spring Boot
  - DBMS: PostgreSQL/MySQL
- È possibile sostituire singole tecnologie nel tempo senza modificare l'intera piattaforma.

## Svantaggi

### Maggiore complessità iniziale

- Progettazione, distribuzione e coordinamento tra livelli richiedono più tempo rispetto a un'architettura monolitica.

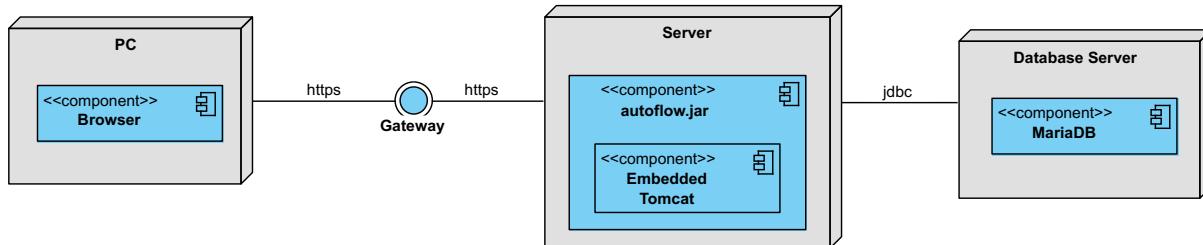
### Overhead di comunicazione

- Le interazioni tra UI, backend e database introdurranno latenza, soprattutto in presenza di numerose richieste simultanee.

### Costi infrastrutturali superiori

- La separazione dei livelli implica più server (o VM), aumentando i costi operativi, specialmente in configurazioni cloud.

*Deploy diagram con sottosistema di orientamento in esecuzione nel nodo del web server.*



### **3.4 Gestione dati persistenti**

La gestione della persistenza nel progetto sarà realizzata utilizzando il database relazionale MariaDB (MySQL), pienamente coerente con i requisiti tecnici del sistema AutoFlow e con le necessità di gestire in modo affidabile tutti i dati relativi allo showroom, alle trattative e alla documentazione di vendita. Tutte le entità del sistema, come Veicolo, Cliente, Addetto Vendite, Amministratore, Proposta, Configurazione e Fattura, verranno memorizzate nel database relazionale, che garantisce integrità referenziale, consistenza delle transazioni e un'organizzazione strutturata delle informazioni. Questa scelta è motivata dal bisogno di mantenere uno storico completo delle operazioni commerciali, permettere ricerche efficienti sui veicoli e tracciare ogni fase delle trattative di vendita.

MariaDB è stato selezionato come DBMS perché offre affidabilità, ottime prestazioni e flessibilità, oltre a un'elevata compatibilità con l'ecosistema MySQL. Il supporto a transazioni ACID, l'utilizzo di chiavi primarie e relazioni esterne, gli indici ottimizzati e le funzionalità avanzate di query consentono di gestire in modo solido i dati relativi ai veicoli, ai clienti e alle proposte. L'architettura transazionale del database garantisce la consistenza delle informazioni in scenari critici come l'aggiornamento dello stato di un veicolo, la modifica di una proposta o la generazione di una fattura, offrendo sicurezza in un ambiente multiutente tipico delle concessionarie.

Per quanto riguarda i documenti generati dal sistema, come preventivi e fatture in formato PDF, essi non verranno salvati direttamente nel database relazionale, ma archiviati nel file system locale, seguendo la struttura definita negli scenari operativi del progetto. AutoFlow organizzerà automaticamente i file in cartelle ordinate per anno e mese, con percorsi come "AutoFlow/ArchivioVendite/Anno/Mese", evitando di sovraccaricare il database con file di grandi dimensioni e garantendo un recupero rapido e ordinato dei documenti amministrativi. Questo approccio permette di alleggerire il carico sul DBMS, mantenendolo dedicato alla gestione dei dati strutturati, e di conservare tutta la documentazione in folder facilmente navigabili, rispecchiando fedelmente le procedure amministrative delle concessionarie.

L'integrazione tra applicazione, archivio dei PDF e database verrà gestita attraverso configurazioni centralizzate in Spring Boot, assicurando comunicazioni sicure, prestazioni ottimizzate e un'interazione coerente tra i vari livelli della piattaforma. La combinazione del database MariaDB per i dati strutturati e dell'archiviazione su filesystem per i documenti generati rappresenta una soluzione solida, scalabile e pienamente adeguata alle esigenze operative del progetto AutoFlow.

### **3.5 Controllo accessi e sicurezza**

La sicurezza dell'applicazione sarà garantita da una serie di misure avanzate che proteggono sia il front-end che il back-end, assicurando una protezione completa dei dati e delle comunicazioni. Il front-end utilizzerà React insieme a un sistema di routing interno dedicato, che gestisce la navigazione tra le varie sezioni della piattaforma, permettendo un'esperienza fluida e controllata sia per i clienti sia per gli addetti alla vendita e gli amministratori. La comunicazione tra client e server avverrà tramite HTTPS, garantendo che tutti i dati trasmessi siano cifrati e protetti da intercettazioni, grazie all'utilizzo del protocollo TLS 1.3 che assicura la riservatezza e l'integrità delle informazioni in transito.

La sicurezza sarà ulteriormente rafforzata dall'utilizzo di una Security Filter Chain che gestirà tutte le richieste in entrata verso il back-end, assicurando che solo quelle correttamente autorizzate possano accedere alle risorse della piattaforma. La Security Filter Chain, basata su Spring Security, permetterà di applicare politiche di autenticazione e autorizzazione in linea con i ruoli definiti dal sistema, garantendo che ogni utente – cliente, addetto vendite o amministratore – possa accedere esclusivamente alle funzionalità di propria competenza.

L'autenticazione sarà inizialmente gestita tramite sessione server-side e cookie sicuri, in modo da semplificare l'implementazione della prima versione della piattaforma. Ogni accesso richiederà l'inserimento delle credenziali, costituite da e-mail e password, e sarà verificato dal sistema tramite controlli interni che determinano la validità dell'account e l'assenza di blocchi. Le password saranno protette tramite hashing irreversibile basato su algoritmi robusti come SHA-512, resistenti ad attacchi di forza bruta e utilizzati per garantire che i dati sensibili non vengano mai memorizzati in chiaro.

Gli addetti vendite creati dall'amministratore riceveranno una password temporanea che dovrà essere cambiata obbligatoriamente al primo accesso, in modo da assicurare che ogni account operativo sia associato a credenziali personali. Il sistema consentirà inoltre la procedura di recupero della password tramite e-mail con token temporaneo e pagina dedicata alla reimpostazione.

Ogni ruolo presente nella piattaforma disporrà di permessi specifici: il cliente potrà accedere allo showroom virtuale, configurare un veicolo e consultare le proprie proposte; l'addetto vendite potrà gestire i veicoli, creare o modificare proposte, seguire trattative in sede e generare fatture; l'amministratore potrà gestire tutti gli account interni, supervisionare le attività e mantenere il controllo generale del sistema. Questa suddivisione garantisce una separazione rigorosa dei privilegi, evitando accessi indesiderati e rispettando il principio del minimo privilegio.

Il sistema prevede infine l'invalidazione automatica della sessione in caso di inattività prolungata, con visualizzazione di un messaggio di sessione scaduta e reindirizzamento alla pagina di login. Questa funzionalità contribuisce a prevenire utilizzi non autorizzati della piattaforma su dispositivi condivisi e si integra perfettamente nelle politiche di sicurezza complessive di AutoFlow.

## Matrice degli accessi

Oggetti\Attori	Utente non autenticato	Cliente	Addetto vendite	Amministratore
Vehicle	listPublicVehicles getVehicleDetails	getVehicleDetails configureVehicle	createVehicle updateVehicle duplicateVehicle setVehicleStatus setPrice assignLicensePlate	createVehicle updateVehicle deleteVehicle duplicateVehicle setVehicleStatus setPrice assignLicensePlate manageVehicleCategories
Configuration	-	createConfiguration getConfiguration updateConfiguration deleteConfiguration	createConfigurationForCustomer getCustomerConfigurations updateConfiguration	createConfigurationForCustomer getAllConfigurations forceDeleteConfiguration
Proposal	-	createProposal getProposals getProposalDetails acceptProposal rejectProposal	createProposalForCustomer getAssignedProposals getProposalDetails updateProposal setProposalDiscount changeProposalStatus cancelProposal	getAllProposals reassignProposal updateProposal forceCancelProposal
CustomerAccount	registerCustomer requestPasswordReset	getProfile updateProfile changePassword getProposals getInvoices requestAccountDeletion	searchCustomer getCustomerProfile updateCustomerContacts getCustomerProposals getCustomerInvoices	createCustomer searchCustomer updateCustomerProfile deactivateCustomer reactivateCustomer resetCustomerPassword

StaffAccount	-	-	getOwnProfile updateOwnProfile changePassword	createStaffUser getStaffUsers updateStaffUser changeStaffRole disableStaffUser enableStaffUser resetStaffPassword
Invoice	-	getInvoices getInvoicePDF	generateInvoiceFromProposal getCustomerInvoices getInvoicePDF	generateInvoiceFromProposal getAllInvoices getInvoicePDF cancelInvoice
StatisticsDashboard	-	-	getDashboard()	getDashboard()

### **3.6 Controllo globale del software**

AutoFlow adotterà un modello di gestione basato su un flusso guidato dagli eventi, nel quale ogni azione effettuata dagli utenti o dai moduli interni genera un evento che attiva la corrispondente logica applicativa. Questo approccio permette al sistema di mantenere un'elevata flessibilità operativa, favorendo una crescita progressiva e una futura distribuzione su più nodi senza richiedere modifiche sostanziali all'architettura. Ogni istanza del software eseguita su un singolo server funzionerà in modo completamente autonomo, senza dipendere dal comportamento delle altre, rendendo l'infrastruttura più semplice da gestire e pronta ad accogliere in seguito configurazioni basate su più server. Qualora AutoFlow venisse distribuito su più nodi, la richiesta dell'utente potrebbe essere inoltrata automaticamente verso il nodo più idoneo tramite meccanismi di distribuzione del carico, così da migliorare l'utilizzo delle risorse e mantenere prestazioni ottimali.

Il web server rimarrà costantemente in ascolto delle richieste provenienti dal browser, che potranno riguardare funzioni applicative (come la gestione dei veicoli, delle configurazioni o delle trattative) oppure risorse statiche del front-end. L'esecuzione dell'applicazione tramite Spring Boot con Tomcat integrato consentirà di gestire sia contenuti statici sia operazioni dinamiche collegate ai controller REST.

Tutte le richieste HTTP verranno prese in carico dalla Dispatcher Servlet, che provvederà a indirizzarle verso il controller appropriato in base all'endpoint richiesto. Per consentire l'elaborazione simultanea di più richieste, il container sfrutterà un sistema di thread pool, che permette di distribuire il lavoro su diversi thread e garantire prestazioni costanti anche in presenza di molte operazioni concorrenti. Grazie a questo meccanismo AutoFlow riuscirà a mantenere tempi di risposta adeguati anche quando il sistema è sottoposto a un carico più elevato, evitando rallentamenti percepibili da parte di clienti, addetti vendite e amministratori. Per assicurare il corretto funzionamento del sistema e la coerenza dei dati, sarà necessario ridurre al minimo i potenziali conflitti tra operazioni simultanee. Tuttavia, alcune funzioni della piattaforma richiederanno comunque una forma di controllo sulle operazioni concorrenti. Nelle prime versioni di AutoFlow sarà utilizzata una strategia basata su lock applicati dal database durante le attività più delicate, come l'aggiornamento dello stato di un veicolo, la modifica di una proposta durante la trattativa o la creazione di un documento amministrativo. Sebbene questo meccanismo sia relativamente semplice, risulta efficace nel garantire che i dati non vengano alterati in modo incoerente quando più utenti accedono alle stesse informazioni.

In una fase successiva, AutoFlow potrà introdurre un'infrastruttura più avanzata basata sulla comunicazione asincrona tramite code di messaggi, particolarmente utile per coordinare processi che coinvolgono simultaneamente più componenti del sistema o che possono essere gestiti in background senza bloccare il flusso principale (come l'archiviazione periodica dei PDF o l'invio di notifiche interne). L'aggiunta di un livello asincrono consentirà di migliorare ulteriormente la scalabilità e la stabilità generale del sistema. Un'ulteriore criticità potrebbe derivare dalla possibilità che uno stesso utente – ad esempio un cliente registrato o un addetto alle vendite – acceda contemporaneamente da più dispositivi o da sessioni diverse. In questi casi non verrà applicata una sincronizzazione diretta tra le sessioni attive, ma la coerenza dei dati sarà comunque garantita dai vincoli ACID del database, che consentiranno solo l'esecuzione delle operazioni compatibili con lo stato attuale delle informazioni. Tutte le operazioni che violano vincoli come l'unicità di un campo o la consistenza dello stato verranno automaticamente rifiutate, assicurando che non si verifichino modifiche contraddittorie e mantenendo quindi l'integrità complessiva del sistema.

### 3.7 Condizioni di boundary

Vengono ora riportate le condizioni di boundary del sistema AutoFlow. La condizione di terminazione è omessa, poiché non sono previsti software specifici per la gestione di questa fase, che verrà trattata tramite processi standard di spegnimento e riavvio del sistema.

#### Configurazione

L'amministratore dovrà predisporre l'ambiente affinché AutoFlow sia accessibile e sicuro. È necessario installare il certificato SSL per permettere al sistema di funzionare tramite protocollo HTTPS e, qualora si utilizzi un dominio personalizzato, sarà importante configurare correttamente i record DNS presso il provider scelto. Le credenziali del database utilizzate in fase di test dovranno essere sostituite con quelle di produzione, così da garantire sicurezza e isolamento dei dati reali. Inoltre, l'amministratore dovrà assicurarsi che sulla macchina siano presenti le cartelle dedicate all'archiviazione dei documenti generati dal sistema, come fatture, preventivi e proposte, poiché AutoFlow utilizza lo storage locale per conservare tali file. Le porte di accesso, come la 80 e la 443, dovranno essere aperte attraverso il firewall, così da permettere agli utenti di accedere al servizio dal browser.

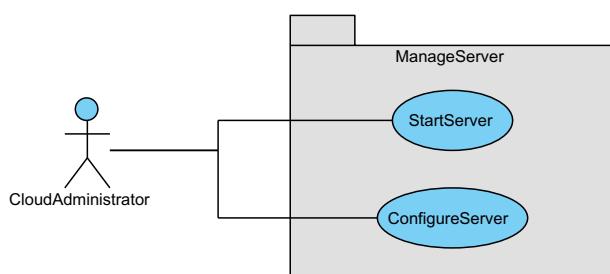
#### Inizializzazione

AutoFlow potrà essere avviato tramite una semplice applicazione Java dedicata, sviluppata appositamente per consentire l'avvio dell'intero sistema in modo immediato e automatizzato. L'amministratore potrà quindi eseguire questa app Java per lanciare AutoFlow, evitando configurazioni complesse. In alternativa, il software potrà essere avviato tramite un servizio di sistema già configurato o, nel caso si utilizzi Docker, attraverso il container predisposto. Nelle installazioni che prevedono più nodi, come uno per l'applicazione e uno per il database, ciascun nodo potrà essere avviato separatamente, senza richiedere procedure aggiuntive.

#### Guasto o malfunzionamento

AutoFlow non è in grado di ripristinare automaticamente lo stato precedente dell'applicazione. Se il problema deriva da un errore nell'infrastruttura, da un'interruzione della macchina virtuale o da un bug interno, l'unica operazione possibile sarà il riavvio dell'applicazione o del server. L'applicazione registrerà comunque ogni eccezione nei file di log interni, così da facilitare la diagnosi e il lavoro di correzione da parte degli sviluppatori. Quando l'errore si verifica nel front-end, come nel caso di un dato inserito in modo errato o di una richiesta non valida, l'utente potrà semplicemente ripetere l'operazione senza dover attendere il riavvio del sistema, garantendo una continuità minima dell'esperienza d'uso.

*Diagramma dei casi d'uso dell'amministratore*



## 4. Servizi dei sottosistemi

Di seguito sono descritti i servizi offerti dai sottosistemi del sistema AutoFlow, evidenziando le funzionalità principali che ciascun sottosistema mette a disposizione per garantire il corretto funzionamento e l'interazione tra le diverse componenti del sistema.

### Authentication

Servizio	Descrizione
<b>Login utente</b>	Gestisce l'autenticazione di clienti, addetti vendite e amministratori.
<b>Logout utente</b>	Termina la sessione e riporta l'utente alla home.
<b>Registrazione cliente</b>	Permette a un nuovo cliente di creare un account.
<b>Attivazione account</b>	Verifica il token e attiva l'account del cliente.
<b>Recupero password</b>	Invia il link di reset e gestisce la nuova password.
<b>Gestione tentativi falliti</b>	Monitora i tentativi errati di accesso e mostra messaggi di errore.
<b>Logout automatico per inattività</b>	Disconnette l'utente dopo un periodo di inattività.

### User Interface

Servizio	Descrizione
<b>Navigazione homepage</b>	Mostra l'home pubblica e l'accesso allo showroom.
<b>Showroom virtuale</b>	Visualizza i veicoli con ricerca e filtri.
<b>Dashboard cliente</b>	Mostra a un cliente configurazioni, proposte e documenti.
<b>Dashboard staff</b>	Mostra KPI, proposte in attesa e strumenti per lo staff.
<b>Form di inserimento/modifica</b>	Fornisce i moduli per creare o modificare dati.
<b>Visualizzazione dettagli veicolo</b>	Mostra schede complete dei veicoli selezionati.
<b>Visualizzazione messaggi di sistema</b>	Mostra notifiche di errore, successo o conferma.

## User Management

Servizio	Descrizione
Gestione account clienti	Crea, modifica, consulta e disattiva account cliente.
Gestione account staff	Permette all'amministratore di gestire gli account interni.
Gestione ruoli e permessi	Assegna i ruoli (cliente, staff, admin) e relativi permessi.
Reset password staff	Consente all'amministratore di reimpostare password interne.
Cambio password utente	Permette all'utente di modificare la propria password.

## Showroom & Vehicle Management

Servizio	Descrizione
Inserimento nuovo veicolo	Crea una scheda veicolo con dati tecnici e immagini.
Modifica dati veicolo	Consente di aggiornare chilometraggio, prezzo o descrizioni.
Gestione stato veicolo	Aggiorna lo stato (Disponibile, Venduto, Non disponibile).
Duplicazione scheda veicolo	Crea una nuova scheda copiando una esistente.
Assegnazione targa/VIN	Gestisce e verifica l'unicità di targa e VIN.
Ricerca e filtri veicoli	Offre funzioni di ricerca avanzata.

## Configuration

Servizio	Descrizione
<b>Creazione configurazione</b>	Permette al cliente o allo staff di configurare un veicolo.
<b>Modifica configurazione</b>	Aggiorna colori, optional e note.
<b>Calcolo prezzo configurazione</b>	Calcola prezzo base, optional, IVA e totale.
<b>Validazione configurazione</b>	Controlla compatibilità optional e disponibilità.
<b>Associazione configurazione-cliente</b>	Collega la configurazione al cliente o alla trattativa.

## Proposals & Sales

Servizio	Descrizione
<b>Creazione proposta</b>	Genera una proposta commerciale da una configurazione.
<b>Gestione proposte online</b>	Permette allo staff di accettare o rifiutare proposte.
<b>Trattativa in sede</b>	Crea proposte per clienti non registrati in concessionaria.
<b>Modifica proposta</b>	Aggiorna optional o condizioni economiche.
<b>Annullamento proposta</b>	Imposta lo stato “Annullata” e notifica il cliente.
<b>Conferma e finalizzazione</b>	Gestisce la conferma e prepara la fatturazione.

## Document Management & PDF

Servizio	Descrizione
<b>Generazione preventivo PDF</b>	Crea il documento di preventivo stampabile.
<b>Generazione fattura PDF</b>	Produce la fattura con numerazione progressiva.
<b>Archiviazione documenti</b>	Salva automaticamente i PDF in cartelle ordinate.
<b>Consultazione documenti cliente</b>	Rende disponibili preventivi e fatture al cliente.
<b>Gestione errori I/O</b>	Registra eventuali problemi di generazione dei file.

## Statistics

Servizio	Descrizione
<b>Calcolo KPI vendite</b>	Calcola indicatori su vendite, trattative e disponibilità.
<b>Generazione report statistici</b>	Produce report e grafici mensili o giornalieri.
<b>Filtri analitici</b>	Permette analisi per periodo, venditore, modello, ecc.
<b>Gestione stati vuoti</b>	Mostra messaggi quando non ci sono dati disponibili.

## Glossario

Termine	Definizione
<b>JDBC</b>	API Java usata dal backend per comunicare con il database relazionale tramite driver dedicati.
<b>MariaDB</b>	DBMS relazionale scelto da AutoFlow per la gestione dei dati persistenti, compatibile con MySQL.
<b>MySQL</b>	Alternativa compatibile a MariaDB utilizzabile come backend relazionale.
<b>ACID</b>	Proprietà fondamentali delle transazioni dei database relazionali (Atomicità, Consistenza, Isolamento, Durabilità).
<b>SHA-512</b>	Algoritmo crittografico utilizzato per l'hashing delle password nel sistema AutoFlow.
<b>TLS 1.3</b>	Protocollo crittografico utilizzato per proteggere il traffico HTTPS tra client e server.
<b>HTTPS</b>	Protocollo HTTP con cifratura TLS che garantisce comunicazioni sicure.
<b>Sessione Server-side</b>	Meccanismo di autenticazione tradizionale che memorizza informazioni di login sul server.
<b>Token di Attivazione</b>	Codice temporaneo inviato al cliente per l'attivazione dell'account.
<b>Spring Boot</b>	Framework backend utilizzato per implementare l'applicazione e i servizi REST.
<b>Embedded Tomcat</b>	Server web integrato in Spring Boot che esegue l'applicazione AutoFlow.
<b>JAR</b>	Formato di pacchetto eseguibile utilizzato per distribuire AutoFlow su server Linux.
<b>WAR</b>	Formato alternativo di distribuzione utilizzabile su application server esterni.
<b>REST API</b>	Interfaccia di comunicazione basata su HTTP per lo scambio di dati frontend-backend.
<b>Filesystem Locale</b>	Sistema di directory sul server dove AutoFlow archivia i PDF generati (preventivi, proposte, fatture).
<b>Security Filter Chain</b>	Catena di filtri di Spring Security che gestisce autenticazione, autorizzazione e sicurezza delle richieste.
<b>SQL Injection</b>	Attacco informatico che manipola query SQL tramite input malevoli; mitigato tramite validazione lato server.
<b>XSS (Cross-Site Scripting)</b>	Attacco che inserisce script malevoli nel frontend; prevenuto tramite sanitizzazione dell'input.