

Università degli Studi di Salerno

Corso di Ingegneria del Software



SDD

System Design

Document

Car – Zone

Versione 0.5

21/11/2024

Coordinatore del progetto:

Nome	Matricola
Francesco Pio Cataudo	0512116773

Partecipanti:

Nome	Matricola
Francesco Pio Cataudo	0512116773
Francesco Santoro	0512117079
Francesco Pio Bottaro	0512118180
Errico Aquino	0512117730

Revision History

Data	Versione	Descrizione	Autori
19/11/2024	0.1	Strutturazione documento ed informazioni iniziali.	Team
20/11/2024	0.2	Stesura decomposizione sottosistemi e mapping hardware-software.	Team
20/11/2024	0.3	Stesura gestioni dati persistenti e controllo degli accessi e sicurezza.	Team
21/11/2024	0.4	Stesura condizione flusso globale del software, boundary condiction, glossario servizi sottosistemi.	Team
21/11/2024	0.5	Revisione generale	Team

Sommario

Revision History.....	3
1.Introduzione	5
1.1 Scopo del Sistema	5
1.2 Design Goals	6
1.3 Definizioni, acronimi, e abbreviazioni.....	9
1.4 Riferimenti	10
1.5 Overview	10
2. Architettura Software Attuale.....	11
3. Architettura Sistema Proposto	11
3.1 Overview	11
3.2 Decomposizione In Sottosistemi.....	12
3.3 Mapping Hardware/Software	15
3.4 Gestione Dati Persistenti	17
3.5 Controlli accessi e sicurezza	19
3.6 Controllo Flusso Globale Software.....	20
3.7 Condizione Limite	21
4 Servizi Dei Sottosistemi	24
5 Glossario.....	27

1.Introduzione

1.1 Scopo del Sistema

Car-Zone è una piattaforma online progettata per rivoluzionare l'esperienza di acquisto di autovetture, semplificando e ottimizzando il processo per clienti e concessionaria. L'obiettivo principale è offrire agli utenti la possibilità di esplorare, selezionare e acquistare un'auto direttamente online, eliminando la necessità di recarsi fisicamente in concessionaria. Questo sistema è pensato per rendere più efficienti tutte le fasi di vendita, riducendo il tempo e le risorse necessarie sia per i clienti che per l'azienda.

La piattaforma Car-Zone offre diverse funzionalità per rispondere alle esigenze di due gruppi principali di utenti: i clienti e gli amministratori.

Per i clienti, Car-Zone consente di registrarsi, effettuare il login, esplorare il catalogo delle auto disponibili, gestire ordini e visualizzare i dettagli delle transazioni. L'interfaccia utente è intuitiva e accessibile, consentendo una navigazione fluida tra le sezioni, facilitando la ricerca dei modelli desiderati e semplificando la finalizzazione degli acquisti.

Dal lato amministrativo, il sistema permette una gestione centralizzata e agevole del catalogo delle autovetture, con la possibilità di aggiungere nuovi modelli, modificarne le caratteristiche e gestire lo stato degli ordini. Gli amministratori possono monitorare le transazioni e visualizzare lo storico degli ordini, assicurando un controllo completo su tutte le operazioni effettuate dai clienti.

Per garantire un'esperienza di utilizzo ottimale, Car-Zone è costruito per essere affidabile, sicuro e scalabile, rispettando specifici requisiti non funzionali come l'accesso simultaneo da parte di più utenti, la sicurezza dei dati personali e le prestazioni rapide. Questo sistema è quindi pensato per supportare la crescita futura dell'azienda, integrando una gestione agile degli ordini e offrendo ai clienti un'esperienza d'acquisto comoda, veloce e moderna.

1.2 Design Goals

I Design Goals sono organizzati in base a diversi criteri: prestazioni, affidabilità, costi, manutenzione e criteri dell'utente finale.

Performance criteria

- **Tempo di risposta:** Car-Zone deve rispondere alle richieste dell'utente entro 5 secondi. Questo assicura un'esperienza utente rapida e reattiva, fondamentale per accessi frequenti al database e per la fruibilità del sito.
- **Throughput:** Il sistema deve essere in grado di gestire simultaneamente più richieste da diversi utenti, specialmente in momenti di traffico elevato. Sarà ottimizzato per processare molteplici richieste simultaneamente, riducendo al minimo i rallentamenti.
- **Uso della memoria:** Il sistema utilizzerà efficientemente la memoria, per bilanciare velocità e consumo di risorse.

Dependability criteria

- **Robustezza:** Car-Zone deve essere in grado di gestire input non validi senza causare interruzioni o comportamenti inaspettati. Verranno implementati controlli di validazione per prevenire crash e rispondere in modo sicuro agli input non corretti o fuori dai limiti.
- **Disponibilità:** Car-Zone deve garantire un'elevata disponibilità, assicurando che il sistema sia accessibile agli utenti per la maggior parte del tempo. L'obiettivo è mantenere una percentuale di uptime che permetta l'uso continuo per le attività quotidiane, minimizzando le interruzioni programmate per la manutenzione e riducendo al minimo i tempi di inattività non previsti.
- **Affidabilità:** Car-Zone deve mantenere un comportamento coerente e affidabile, con output che rispecchino correttamente le aspettative dell'utente anche in presenza di errori imprevisti.
- **Tolleranza ai guasti:** Il sistema deve continuare a funzionare in condizioni di errore (ad esempio, perdita temporanea di connettività).
- **Sicurezza:** Poiché Car-Zone gestirà dati sensibili degli utenti, il sistema sarà progettato per resistere ad attacchi malevoli, assicurando che le informazioni siano protette tramite autenticazione e crittografia adeguata.
- **Disponibilità:** Il sistema deve essere disponibile per gli utenti il più possibile, con una strategia di manutenzione programmata per minimizzare i tempi di inattività.

Cost criteria

- **Costo di sviluppo:** I costi iniziali di sviluppo saranno ottimizzati tramite una progettazione modulare e componenti riutilizzabili, riducendo il tempo necessario per future implementazioni e aggiornamenti.
- **Costo di distribuzione:** L'implementazione e l'installazione del sistema devono essere eseguite in modo da minimizzare i costi per gli utenti.
- **Costo di manutenzione:** Il sistema sarà progettato per facilitare correzioni di bug e aggiornamenti, garantendo nel contempo un basso costo di manutenzione a lungo termine.

Maintenance criteria

- **Estensibilità:** Car-Zone sarà progettato in modo modulare per permettere l'aggiunta di nuove funzionalità in futuro, con il minimo impatto sul sistema esistente.
- **Modificabilità:** Le funzionalità di base devono poter essere aggiornate o modificate con facilità, per poter adattare il sistema a nuovi requisiti o funzionalità richieste dal mercato.
- **Adattabilità:** Il sistema deve essere facilmente adattabile a diversi domini applicativi, permettendo il riutilizzo delle funzionalità principali in altri contesti o settori.
- **Portabilità:** Il sistema sarà sviluppato per essere compatibile con diverse piattaforme, così da facilitarne il trasferimento o l'adattamento in altre applicazioni o contesti.
- **Leggibilità:** Il codice del sistema deve essere scritto in modo chiaro e leggibile per facilitare la comprensione da parte degli sviluppatori, sia attuali che futuri.
- **Tracciabilità dei requisiti:** Ogni requisito sarà mappato al codice specifico per facilitare le verifiche e garantire che il sistema soddisfi le specifiche concordate.

End User criteria

- **Utilità:** Car-Zone deve supportare efficientemente le attività chiave dell'utente, come la ricerca e l'interazione con i veicoli disponibili, per garantire una user experience ottimale.
- **Usabilità:** L'interfaccia del sistema sarà intuitiva e facile da usare per utenti di diversi livelli di esperienza. L'accesso e la navigazione saranno ottimizzati per l'uso su desktop e dispositivi mobili, con un design responsive per migliorare l'accessibilità.

I trade-off sono necessari per bilanciare i design goals con le limitazioni di risorse, tempo, e budget.

- **Tempo di risposta vs. Consumo di memoria**

Per migliorare la velocità del sistema, verrà utilizzata una cache mirata che bilancia le prestazioni e il consumo di memoria, memorizzando solo le informazioni più utili per gli utenti.

- **Costo di sviluppo vs. Qualità**

Per ottimizzare i costi senza sacrificare troppo la qualità, si darà priorità alle funzionalità principali, rinviando quelle meno urgenti e garantendo il rilascio puntuale del sistema.

- **Sicurezza vs. Usabilità**

La sicurezza verrà progettata per essere efficace ma semplice, con soluzioni come l'autenticazione a due fattori opzionale, che proteggono i dati senza complicare eccessivamente l'esperienza utente.

- **Scalabilità vs. Costi di manutenzione**

Si adotterà un'architettura modulare dove necessario, implementando miglioramenti nelle altre aree solo in base a esigenze future, così da controllare i costi di manutenzione.

- **Tempo di consegna vs. Funzionalità**

Per rispettare le scadenze, verranno implementate prima le funzionalità essenziali, lasciando quelle secondarie per aggiornamenti successivi, garantendo così un rilascio nei tempi stabiliti.

1.3 Definizioni, acronimi, e abbreviazioni

Definitions

- **Cliente:** Utente che utilizza la piattaforma per esplorare, selezionare e acquistare autovetture. Include dati personali quali id cliente, nome, cognome, e-mail, password, genere, data di nascita e indirizzo.
- **Amministratore:** Utente con privilegi avanzati per la gestione delle entità del sistema (ad esempio, autovetture e ordini). Include informazioni personali come id amministratore, nome, cognome, e-mail e password.
- **Utente Registrato:** Qualsiasi utente che dispone di un account sulla piattaforma. Può essere un cliente o un amministratore.
- **Utente Non Registrato:** Persona che utilizza la piattaforma senza registrarsi. Può accedere solo a funzionalità limitate, come la visualizzazione del catalogo.
- **Autovettura:** Rappresentazione di un veicolo disponibile nel sistema, con informazioni quali id automobile, tipo di auto, marca, modello, caratteristiche tecniche e descrizione.
- **Ordine:** Acquisto confermato da un cliente e approvato da un amministratore, con dati come id ordine, date rilevanti, ricevuta d'acquisto e data minima di ritiro.
- **Conferma Ordine:** Azione eseguita dal cliente per confermare l'acquisto di un'autovettura.
- **Approvazione Ordine:** Processo eseguito dall'amministratore per convalidare un ordine.

Acronimi:

- **MVC:** Model-View-Controller, un pattern architetturale che separa i dati (Model), la logica di business (Controller) e la presentazione (View).
- **CRUD:** Create, Read, Update, Delete; operazioni fondamentali per la gestione di dati in un database.
- **DBMS:** Database Management System, sistema utilizzato per gestire i dati persistenti.
- **HTTP/HTTPS:** HyperText Transfer Protocol/Secure, protocolli per la comunicazione tra client e server.
- **ACL:** Access Control List, elenco di regole per gestire i permessi di accesso alle risorse.

Abbreviazioni

- **ID:** Identifier, identificativo univoco utilizzato per distinguere entità come utenti, auto e ordini.
- **UI/UX:** User Interface/User Experience, elementi di design e funzionalità per l'interazione utente.
- **SQL:** Structured Query Language, linguaggio per gestire i dati nei database relazionali.

1.4 Riferimenti

- Documento di Analisi dei requisiti relativo a questo progetto
- Object-oriented-Software-Engineering-3rd-Edition

1.5 Overview

Il documento è organizzato nel seguente modo:

1. **Introduzione:** Definisce il contesto e gli obiettivi del sistema, evidenziando l'importanza di un design scalabile, sicuro ed efficiente per trasformare il processo di vendita di autovetture in un'esperienza digitale.
2. **Architettura del Sistema Attuale:**
Analizza il sistema esistente, basato su processi manuali e privo di strumenti digitali, sottolineandone le inefficienze e le limitazioni operative.
3. **Architettura del Sistema Proposto:**
Fornisce una visione d'insieme del nuovo sistema, includendo:
 - Decomposizione in Sottosistemi e interazioni.
 - Mapping Hardware/Software
 - Gestione Dati Persistenti
 - Controlli accessi e sicurezza
 - Controllo Flusso Globale Software
 - Condizione Limite
 - Servizi dei Sottosistemi
4. **Servizi dei Sottosistemi:** rappresentati i servizi dei sottosistemi.
Glossario: Raccolta di vocabali non comuni utilizzati nel documento

2. Architettura Software Attuale

Attualmente, il sistema utilizzato da Car-Zone si basa interamente su interazioni manuali e non dispone di una piattaforma online per la gestione delle vendite. Questo comporta che il cliente, interessato all'acquisto di un'auto, debba contattare la concessionaria tramite telefono o email per ottenere informazioni preliminari sulla disponibilità delle auto. Successivamente, è necessario fissare un appuntamento e recarsi fisicamente presso la concessionaria per consultare il catalogo delle auto e procedere all'acquisto.

Durante il processo, non esiste un sistema centralizzato per il tracciamento degli ordini. Le informazioni vengono comunicate al cliente tramite chiamate o email, mentre lo stato dell'ordine e le date di consegna sono gestiti manualmente. Infine, il cliente deve ritornare in concessionaria alla data prestabilita per ritirare il veicolo.

Questo approccio presenta diverse limitazioni. L'assenza di un sistema digitale rallenta il processo e richiede un notevole impegno di tempo da parte sia del cliente che dell'azienda. Inoltre, la gestione manuale delle operazioni aumenta il rischio di errori e rende il processo meno efficiente. Il catalogo dei veicoli non è accessibile online e le sue informazioni non sono aggiornate in tempo reale, limitando la visibilità delle offerte. L'accessibilità per i clienti è ridotta, poiché richiede necessariamente la presenza fisica in concessionaria.

3. Architettura Sistema Proposto

3.1 Overview

Il sistema proposto per Car-Zone si basa su un'architettura moderna organizzata secondo il pattern MVC (Model-View-Controller), che permette di separare nettamente la gestione dei dati, la logica applicativa e la presentazione. Questo approccio consente di superare le limitazioni del sistema attuale, garantendo maggiore efficienza, scalabilità e manutenibilità.

Le funzionalità offerte variano in base al tipo di utente:

- **I clienti registrati** possono esplorare il catalogo, visualizzare lo storico ordini e finalizzare gli acquisti direttamente online.
- **Gli amministratori** possono aggiungere, eliminare o modificare i veicoli nel catalogo, approvare gli ordini e visualizzare lo storico dei clienti e degli ordini.

- **Gli utenti non registrati** hanno accesso limitato al sistema, potendo consultare solo il catalogo delle auto e visualizzarle in modo dettagliato. Tuttavia, possono registrarsi per accedere alle funzionalità complete.

L'adozione di un'architettura MVC consente di separare chiaramente le responsabilità, migliorando la modularità del sistema. Ad esempio, eventuali modifiche al layout dell'interfaccia utente non richiedono interventi sul database o sulla logica di business.

3.2 Decomposizione In Sottosistemi

Identificazione dei sottosistemi per CarZone

1. Sottosistema di Autenticazione e Sicurezza

- **Descrizione:** Gestisce l'autenticazione degli utenti, il controllo degli accessi e la protezione dei dati sensibili tramite crittografia e altre misure di sicurezza.
- **Funzionalità principali:** Login, registrazione, gestione dei ruoli e autorizzazioni.

2. Sottosistema di Gestione delle Auto

- **Descrizione:** Si occupa di organizzare, memorizzare e aggiornare i dati relativi alle auto disponibili sulla piattaforma.
- **Funzionalità principali:** Aggiunta, modifica, rimozione e visualizzazione dei dettagli delle auto (marca, modello, prezzo, ecc.).

3. Sottosistema di Ricerca e Filtro

- **Descrizione:** Permette agli utenti di cercare e filtrare le auto in base a parametri come marca, modello, e tipo.
- **Funzionalità principali:** Implementazione di query dinamiche per la ricerca e opzioni di filtro avanzate.

4. Sottosistema di Gestione degli Ordini

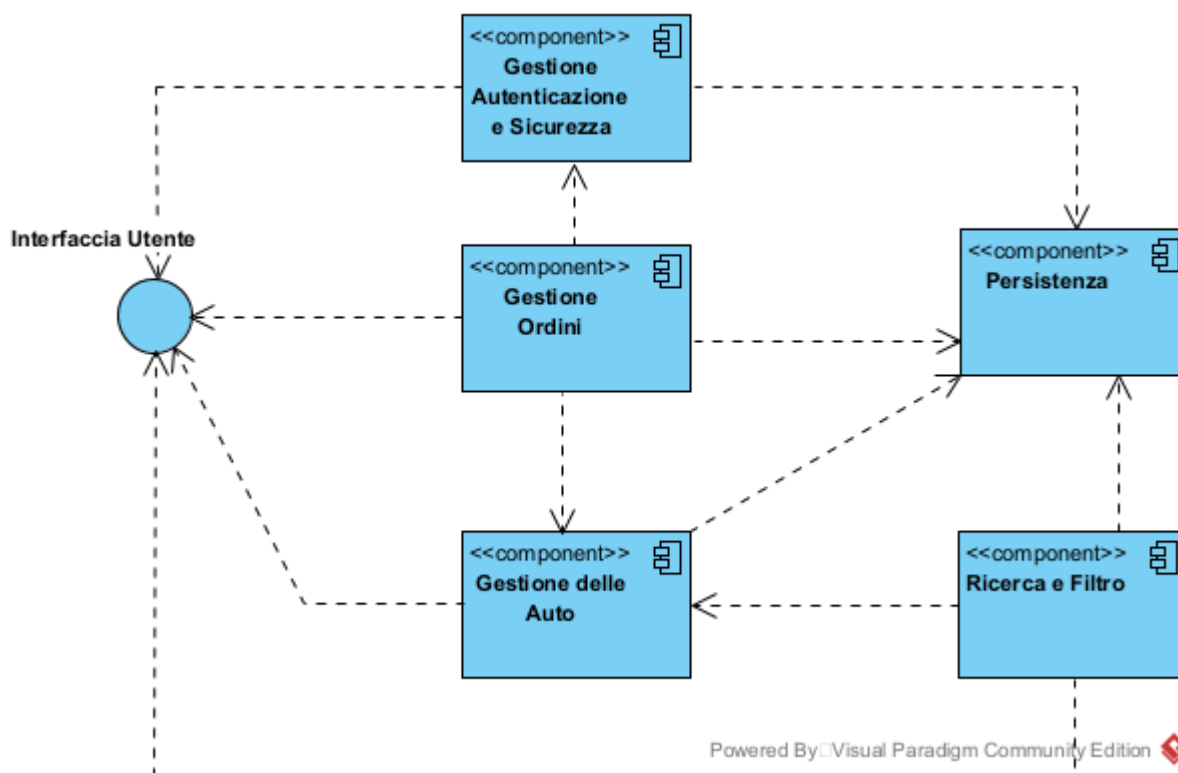
- **Descrizione:** Gestisce ordini delle auto effettuati dai clienti.
- **Funzionalità principali:** Conferma e monitoraggio degli ordini.

5. Sottosistema di Persistenza

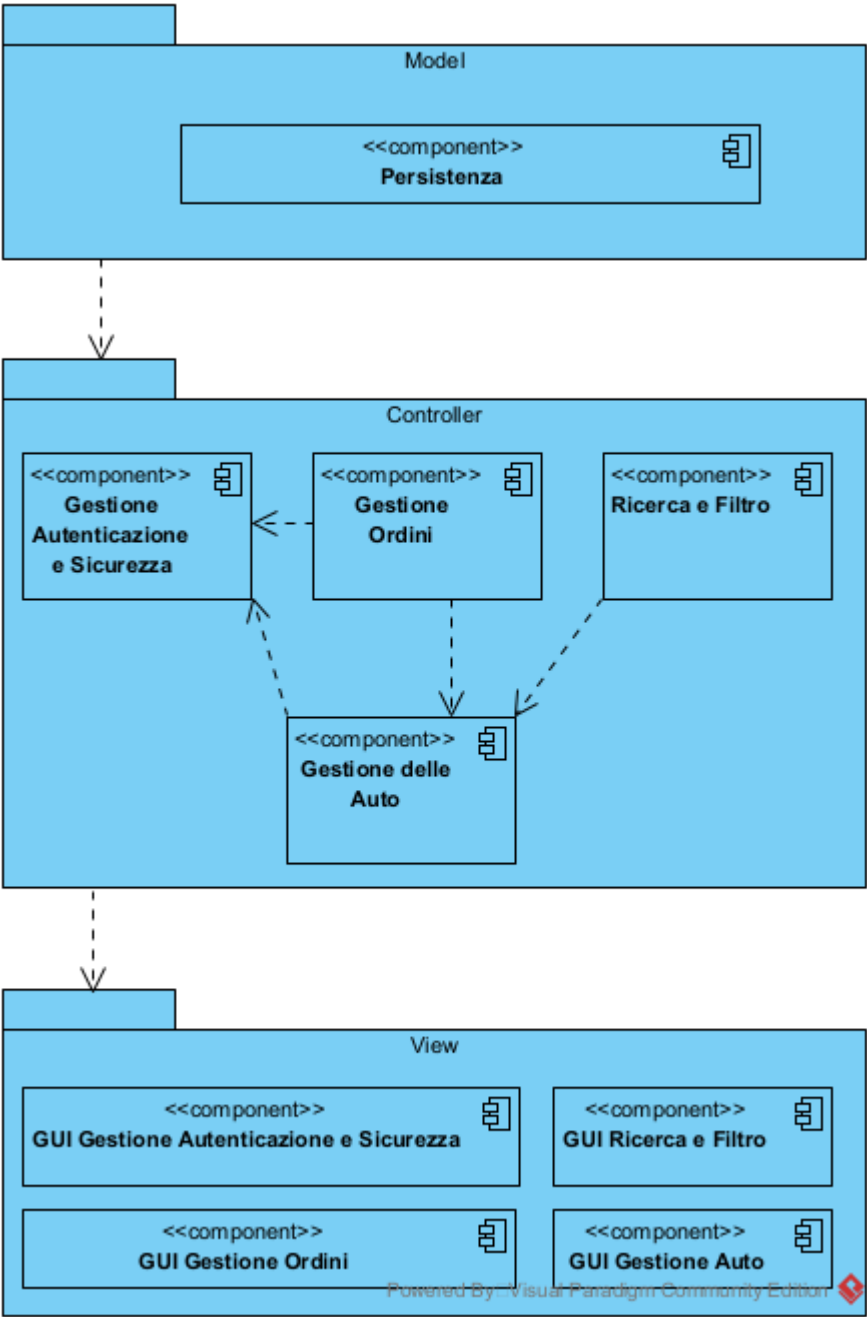
- **Descrizione:** Si occupa dell'archiviazione, recupero e gestione dei dati persistenti del sistema, inclusi utenti, auto e ordini.
- **Funzionalità principali:** Operazioni CRUD (Create, Read, Update, Delete) ottimizzate per garantire tempi di risposta rapidi.

6. Sottosistema di Interfaccia Utente (UI)

- **Descrizione:** Gestisce l'interfaccia utente per assicurare un'esperienza utente intuitiva e user-friendly sia su dispositivi desktop che mobili.
- **Funzionalità principali:** Pagine per la visualizzazione e interazione, feedback sulle azioni dell'utente e supporto per design responsive.



Di seguito sottosistemi organizzati nell'architettura scelta:



3.3 Mapping Hardware/Software

CarZone è un sistema distribuito che opera su una configurazione multi-nodo per gestire sia il carico di lavoro che la distribuzione geografica degli utenti. La mappatura hardware/software prevede tre nodi principali: il Web Server, il Database Server e il Client. Ogni nodo è progettato per ospitare specifici sottosistemi, ottimizzando le prestazioni e la modularità del sistema.

Nodo 1: Web Server

Il Web Server è il cuore del sistema, responsabile della gestione delle richieste degli utenti, dell'elaborazione delle logiche di business e della comunicazione con il database.

- Sottosistemi ospitati:
 - Sottosistema di autenticazione: Gestisce il login, la registrazione e la validazione degli utenti.
 - Sottosistema di Gestione delle Auto: Gestisce le operazioni CRUD (Create, Read, Update, Delete) sulle auto.
 - Sottosistema di Ricerca e Filtro: Elabora le richieste di ricerca e applica filtri ai dati.
 - Sottosistema di Gestione degli Ordini: Gestisce gli ordini.
- Tecnologie:
 - Server Web: Apache.
 - Linguaggio di Programmazione: Java.
 - Protocollo di Comunicazione: HTTP/HTTPS per comunicare con i client.

Nodo 2: Database Server

Il Database Server è dedicato alla gestione e alla conservazione dei dati persistenti del sistema. Include tutti i dati relativi agli utenti, alle auto e agli ordini. Il nodo ospita un sistema di gestione del database (DBMS) come MySQL.

- Sottosistemi ospitati:
 - Persistence Subsystem: Archivia e recupera dati tramite operazioni CRUD standard.
 - Data Synchronization: Garantisce la consistenza dei dati tra il database e i nodi applicativi.
- Tecnologie:
 - DBMS: MySQL.
 - Infrastruttura: Server dedicato con storage ridondante per assicurare affidabilità.

Nodo 3: Client

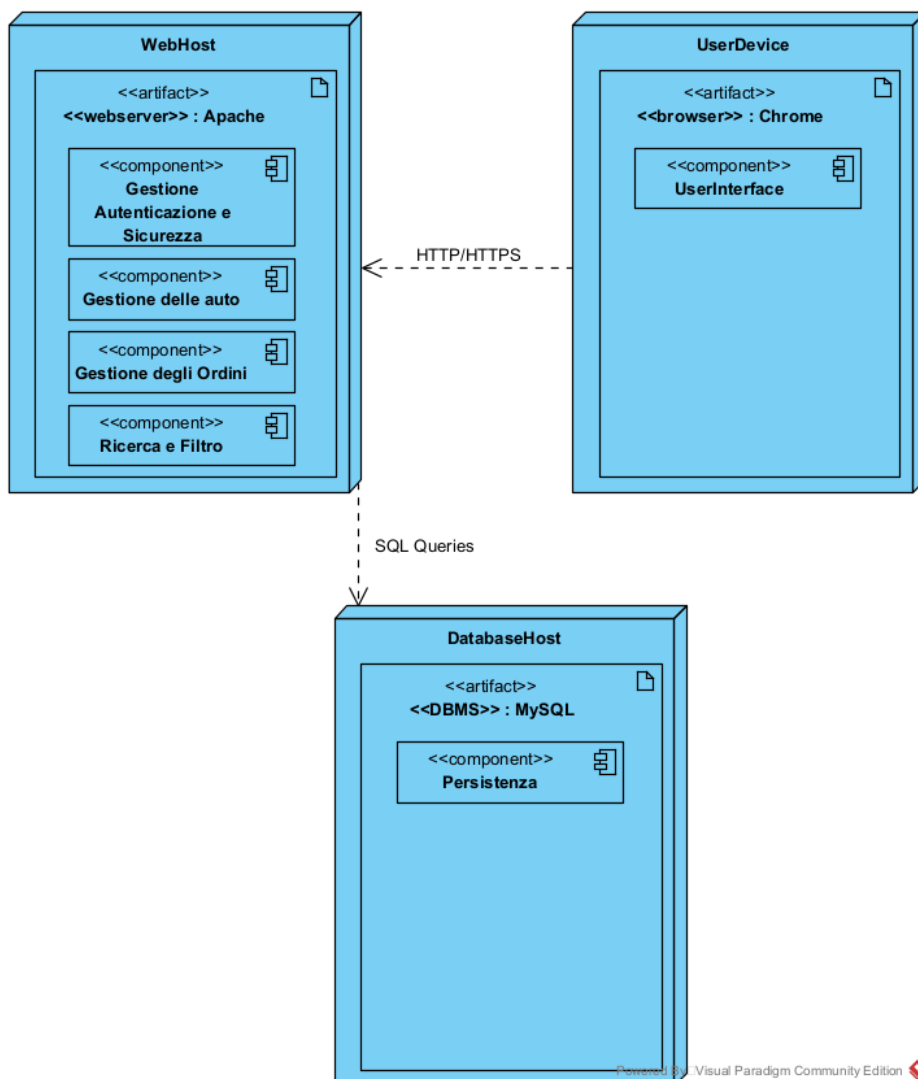
Il Client rappresenta l'interfaccia utente attraverso cui gli utenti interagiscono con CarZone.

- Sottosistemi ospitati:
 1. Sottosistema di Interfaccia Utente: Mostra le informazioni agli utenti e gestisce l'interazione tramite interfacce grafiche intuitive.
- Tecnologie:
 - Browser: Chrome, Safari, Firefox.

Comunicazione tra Nodi

La comunicazione tra i nodi avviene tramite protocolli standard:

- Client-Web Server: Utilizza HTTP/HTTPS per inviare richieste e ricevere risposte.
- Web Server-Database Server: Utilizza query SQL per accedere e manipolare i dati nel database.



3.4 Gestione Dati Persistenti

La gestione dei dati persistenti è fondamentale per garantire che informazioni critiche siano conservate e recuperabili anche in caso di riavvio del sistema o di eventi imprevisti. Nel contesto del progetto, è stato deciso di utilizzare una strategia di gestione della persistenza basata su un Database Relazionale (MySQL) per sfruttare la sua capacità di gestire dati complessi e assicurare scalabilità, integrità e sicurezza.

Identificazione degli Oggetti Persistenti

Gli oggetti persistenti sono stati individuati attraverso l'analisi delle entità chiave che devono sopravvivere alla chiusura del sistema. Questi includono:

- **Utenti:** Informazioni su Admin e Clienti, accorpati in una singola entità per semplificare la gestione.
- **Auto:** Dati relativi alle auto come modello, prezzo, caratteristiche tecniche.
- **Ordini:** Informazioni sugli acquisti effettuati dagli utenti.

Accorpamento di Admin e Cliente

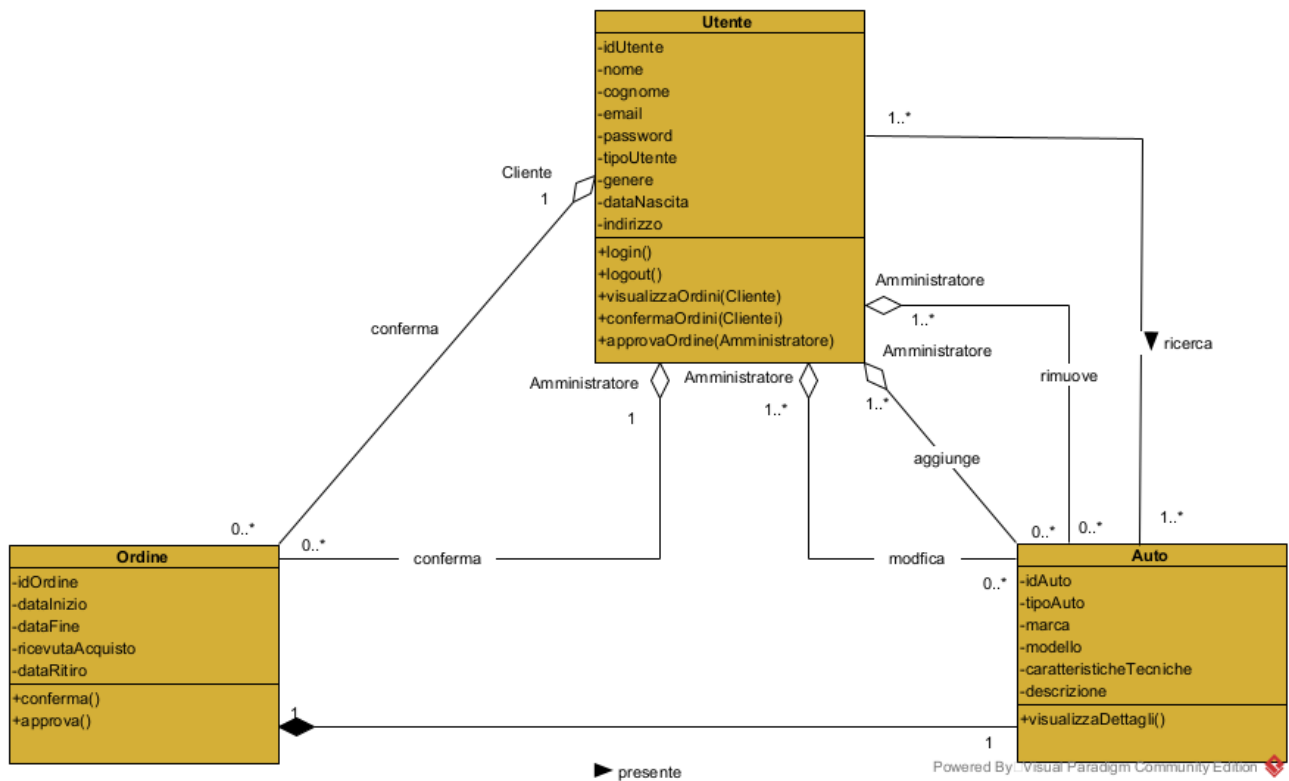
Per semplificare la gestione dei dati, è stato deciso di accorpare gli utenti **Admin** e **Cliente** in un'unica entità **Utente**, differenziandoli tramite un attributo ruolo. Questo approccio offre i seguenti vantaggi:

- Riduce la complessità delle tabelle nel database.
- Consente una gestione uniforme delle operazioni CRUD.
- Semplifica la gestione dei permessi e l'integrazione con i moduli di autenticazione.

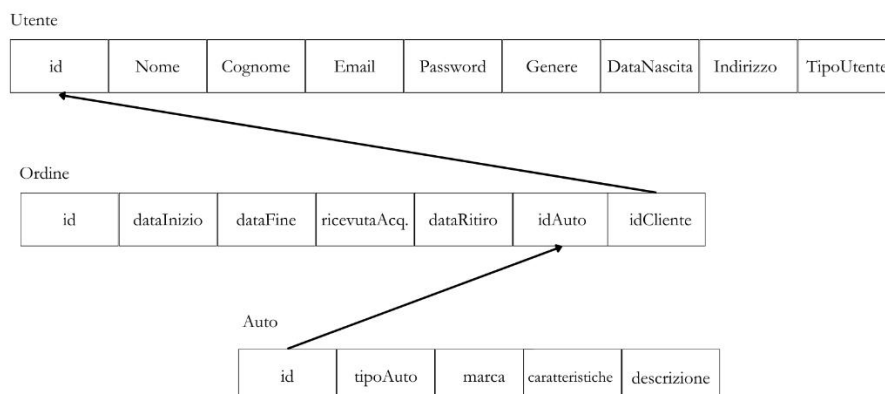
La tabella **Utenti** include:

- **Attributi comuni:** ID, Nome, Cognome, Email, Password.
- **Ruolo:** Specifica se l'utente è un **Admin** o un **Cliente**.
- **Informazioni aggiuntive:** Per esempio, il Cliente può avere uno storico degli ordini, mentre l'Admin ha accesso a privilegi gestionali.

Di seguito il Class Diagramm con accorpamento di Admin e Cliente in un'unica entità Utente:



Di seguito schema logico del sistema:



3.5 Controlli accessi e sicurezza

Matrice di Controllo degli Accessi (Access Control Matrix)

Oggetti / Attori	Sessione Utente	Auto	Ordine
Cliente	- Login per accedere al sistema. - Logout per terminare la sessione attiva.	- Visualizza il catalogo di autovetture	- Crea ordini - Visualizza ordini - Confirma ordini
Utente Non Registrato	- Registrazione per creare un account Cliente.	- Visualizza il catalogo di autovetture	- Nessun accesso
Amministratore	- Login per accedere al sistema. - Logout per terminare la sessione attiva.	- Aggiunge, modifica, elimina autovetture	- Visualizza ordini clienti - Approva ordini clienti - Visualizza storico clienti

- **Cliente:**
 - Può effettuare login per accedere al sistema e logout per terminare la sessione attiva.
 - Può visualizzare il catalogo delle autovetture.
 - Può creare, visualizzare i propri ordini e confermarli.
- **Utente Non Registrato:**
 - Può registrarsi per creare un account Cliente.
 - Può visualizzare il catalogo delle autovetture.
 - Non ha accesso alla gestione degli ordini.
- **Amministratore:**
 - Può effettuare login per accedere al sistema e logout per terminare la sessione attiva.
 - Può manipolare il catalogo delle autovetture (può aggiungere, modificare o eliminare auto).

- Può visualizzare tutti gli ordini dei clienti e approvarli, inoltre ha la possibilità di visualizzare lo storico dei clienti.

3.6 Controllo Flusso Globale Software

Il sistema CarZone adotta un paradigma di controllo del flusso event-driven (basato su eventi) per gestire le richieste degli utenti e le operazioni di sistema. Questo approccio consente al sistema di rispondere in tempo reale agli eventi generati dagli utenti (come la visualizzazione delle autovetture, la gestione degli ordini e l'autenticazione), migliorando la reattività e il throughput del sistema. L'utilizzo di thread separati per ogni richiesta garantisce un'elaborazione parallela delle operazioni, ottimizzando le prestazioni del sistema. Il sistema è progettato per rispondere a ogni evento (ad esempio, una richiesta HTTP del cliente) non appena si verifica. Gli eventi possono essere richieste utente, risposte del sistema, o modifiche ai dati.

3.7 Condizione Limite

Le boundary conditions (condizioni di limite) rappresentano le condizioni critiche di avvio, arresto e gestione di eventi eccezionali nel sistema CarZone. L'obiettivo principale di queste condizioni è garantire la resilienza, affidabilità e continuità operativa del sistema, gestendo in modo efficace le problematiche legate ai guasti hardware, software, interruzioni di rete e altre anomalie che potrebbero compromettere il corretto funzionamento del sistema.

Configurazione

Nel sistema Car-Zone, alcune entità persistenti, come autovetture e ordini, necessitano di configurazioni specifiche per il loro corretto utilizzo. Questi scenari comprendono:

- Inizializzazione del catalogo autovetture:
 - Il catalogo viene popolato al momento del caricamento iniziale dei dati o aggiornato manualmente dall'Amministratore.
- Gestione degli ordini:
 - Gli ordini vengono archiviati e recuperati per garantire la continuità operativa, anche in caso di interruzioni del sistema.

Use Case di Configurazione

- Gestione Catalogo Autovetture:
 - Aggiunta, modifica ed eliminazione delle autovetture tramite il sottosistema di Gestione Autovetture.
- Gestione Persistenza Ordini:
 - Validazione e archiviazione degli ordini tramite il sottosistema di Gestione Ordini, con meccanismi per rilevare dati corrotti e ripristinare la consistenza.

Avvio e Arresto

Use Case di Avvio e Spegnimento

Per garantire la stabilità e l'affidabilità del sistema, sono stati definiti scenari per l'avvio e l'arresto sicuro dei sottosistemi.

1. Avvio del Sistema:
 - Inizializza i sottosistemi principali:
 - Gestione Autovetture: Caricamento del catalogo.
 - Gestione Ordini: Caricamento degli ordini esistenti.
 - Verifica la consistenza dei dati persistenti, riparando eventuali corruzioni.

2. Arresto del Sistema:

- Chiude in modo sicuro tutti i sottosistemi, garantendo che i dati persistenti siano salvati correttamente.
- Genera un log di sistema per tracciare lo stato del sistema al momento dell'arresto.

Gestione delle Eccezioni

Le principali fonti di errore nel sistema Car-Zone includono:

1. Hardware failure:

- Problemi: Guasti al disco rigido o interruzioni di rete.
- Soluzioni:
 - Ripristino automatico al riavvio.

2. Cambiamenti nell'ambiente operativo:

- Problemi: Interruzioni di rete o blackout.
- Soluzioni:
 - Gestione dello stato temporaneo degli ordini.
 - Ripristino automatico delle attività al ritorno delle condizioni operative.

3. Errori software:

- Problemi: Bug o incoerenze nei dati tra i sottosistemi.
- Soluzioni:
 - Verifiche di consistenza dei dati.
 - Generazione di log di errore.

Use Case Boundary

Use Case: StartServer

Nome del caso d'uso	StartServer
Condizione d'ingresso	L'Amministratore accede al server CarZone tramite credenziali valide.
Sequenza degli eventi	L'Amministratore esegue il comando startCarZoneSystem. Il sistema verifica lo stato precedente del sistema: <ul style="list-style-type: none">• Se il sistema è stato arrestato correttamente, carica il catalogo autovetture e l'elenco ordini attivi.

	<ul style="list-style-type: none"> Se il sistema è stato interrotto bruscamente, notifica l'Amministratore dell'errore e esegue controlli di consistenza sui dati persistenti, riparandoli se necessario.
Condizione d'uscita	Il sistema è avviato correttamente e i sottosistemi sono pronti per ricevere richieste.

Use Case: ShutdownServer

Nome del caso d'uso	ShutdownServer
Condizione d'ingresso	L'Amministratore accede al server CarZone tramite credenziali valide.
Sequenza degli eventi	<p>L'Amministratore esegue il comando shutdownCarZoneSystem.</p> <p>Il sistema chiude gradualmente tutti i sottosistemi:</p> <ul style="list-style-type: none"> Salva i dati attivi del catalogo e archivia gli ordini non completati. Registra i log operativi.
Condizione d'uscita	I sottosistemi sono chiusi in sicurezza e i dati persistenti sono aggiornati e integri.

Use Case: HandleFailure

Nome del caso d'uso	HandleFailure
Condizione d'ingresso	Si verifica un errore nel sistema (hardware, software o rete).
Sequenza degli eventi	<p>Il sistema rileva il tipo di errore:</p> <ul style="list-style-type: none"> Errore hardware: Salva temporaneamente lo stato (se possibile) e notifica l'Amministratore. Errore di rete: Mette in pausa le operazioni attive e le riprende al ripristino della connessione. Errore software: Genera un log di errore e invia notifiche agli utenti.
Condizione d'uscita	L'errore è stato gestito e il sistema è ripristinato con i dati integri.

4 Servizi Dei Sottosistemi

Ogni sottosistema di CarZone fornisce una serie di servizi specifici che permettono al sistema di eseguire operazioni critiche e di interagire in modo efficace con gli utenti e con altri sottosistemi. Di seguito sono identificati i servizi principali per ogni sottosistema.

Sottosistema di Autenticazione e Sicurezza

- **Descrizione:** Gestisce l'autenticazione degli utenti e il controllo degli accessi, assicurando che solo gli utenti autorizzati possano accedere alle risorse e alle informazioni sensibili.

Servizi identificati:

- **Login:** Verifica delle credenziali degli utenti (nome utente e password).
- **Registrazione:** Consente a nuovi utenti di creare un account.
- **Gestione Ruoli e Permessi:** Assegna e verifica i ruoli (Cliente, Amministratore) e i permessi di accesso per ciascun utente.
- **Autorizzazione:** Verifica se l'utente ha il permesso di eseguire un'operazione richiesta in base al suo ruolo.

Sottosistema di Gestione delle Auto

- **Descrizione:** Si occupa della gestione delle informazioni relative alle autovetture disponibili sulla piattaforma, permettendo di aggiungere, modificare, rimuovere e visualizzare le auto nel catalogo.

Servizi identificati:

- **Aggiungi Auto:** Consente agli amministratori di aggiungere nuove autovetture al sistema, con i dettagli come marca, modello, prezzo, e descrizione.
- **Modifica Auto:** Permette di modificare i dettagli delle autovetture esistenti.
- **Rimuovi Auto:** Consente di rimuovere le autovetture dal catalogo.
- **Visualizza Auto:** Permette agli utenti di visualizzare i dettagli delle autovetture presenti nel catalogo.

Sottosistema di Ricerca e Filtro

- Descrizione: Permette agli utenti di cercare e filtrare le autovetture in base a vari criteri, come marca e modello.

Servizi identificati:

- Cerca Auto: Esegue la ricerca nel catalogo delle autovetture in base ai criteri forniti dall'utente (ad **esempio, marca, modello, ecc.**).

Sottosistema di Gestione degli Ordini

- Descrizione: Gestisce gli ordini effettuati dai clienti, compresa la conferma, il monitoraggio e la gestione degli ordini fino al completamento.

Servizi identificati:

- Crea Ordine: Consente ai clienti di effettuare un nuovo ordine per l'acquisto di un'auto.
- Visualizza Ordine: Permette ai clienti di visualizzare lo stato del loro ordine (ad esempio, in attesa, in elaborazione, completato).
- Conferma Ordine: Gestisce la conferma dell'ordine.
- Monitoraggio Ordine: Permette agli utenti di monitorare l'avanzamento dell'ordine.

Sottosistema di Persistenza

- Descrizione: Si occupa della gestione dei dati persistenti, tra cui autovetture, ordini e utenti, garantendo l'archiviazione e il recupero dei dati in modo efficiente e sicuro.

Servizi identificati:

- Salvataggio Dati: Esegue operazioni di salvataggio dei dati relativi a ordini, autovetture, e utenti nel sistema di archiviazione.
- Recupero Dati: Recupera i dati salvati (autovetture, ordini, utenti) per l'utilizzo nelle operazioni successive.
- Operazioni CRUD (Create, Read, Update, Delete): Gestisce le operazioni fondamentali di modifica, visualizzazione, aggiornamento e eliminazione dei dati nel sistema.
- Backup e Ripristino: Gestisce il backup periodico dei dati e il ripristino in caso di guasti del sistema.

Sottosistema di Interfaccia Utente (UI)

- Descrizione: Si occupa della gestione dell'interfaccia utente, fornendo una user experience ottimale su dispositivi desktop e mobili.

Servizi identificati:

- Visualizza Catalogo: Visualizza il catalogo delle autovetture disponibili con tutte le informazioni pertinenti.
- Gestisci Ordini: Permette agli utenti di visualizzare, modificare o cancellare i propri ordini.
- Feedback Utente: Fornisce feedback visivi e interattivi per le azioni dell'utente, come messaggi di conferma o errore.
- Navigazione: Gestisce la navigazione tra le varie pagine del sistema (catalogo, ordine, ricerca, ecc.).
- Design Responsivo: Adatta l'interfaccia in base al dispositivo utilizzato (desktop, tablet, smartphone) per una migliore esperienza utente.

5 Glossario

Termini Generali

- **Cliente:** Utente registrato che utilizza la piattaforma per acquistare autovetture. Ogni cliente ha un account con informazioni personali come ID, nome, e-mail e indirizzo.
- **Amministratore:** Utente con privilegi avanzati per gestire il catalogo delle auto, ordini e altre operazioni amministrative. Ha un account con ID, nome, e-mail e password.
- **Utente Registrato:** Utente (cliente o amministratore) che ha un account sulla piattaforma.
- **Utente Non Registrato:** Persona che utilizza la piattaforma senza registrarsi. Può solo visualizzare il catalogo delle auto.
- **Autovettura:** Oggetto rappresentante un'auto, con informazioni quali ID, marca, modello, caratteristiche tecniche e descrizione.
- **Ordine:** Acquisto confermato di un'auto, comprendente ID, ricevuta, data minima di ritiro e altre informazioni rilevanti.
- **Conferma Ordine:** Azione del cliente per confermare un ordine.
- **Approvazione Ordine:** Azione dell'amministratore per convalidare un ordine.

Acronimi e Abbreviazioni

- **MVC:** (Model-View-Controller) Architettura software che separa la gestione dei dati (Model), la logica applicativa (Controller) e la presentazione (View).
- **CRUD:** (Create, Read, Update, Delete) Operazioni di base per la gestione dei dati.
- **DBMS:** (Database Management System) Sistema software per la gestione di database (ad esempio, MySQL).
- **HTTP/HTTPS:** (HyperText Transfer Protocol/Secure) Protocolli per la comunicazione tra client e server.
- **ACL:** (Access Control List) Elenco di regole per gestire i permessi di accesso alle risorse.
- **ID:** Identifier, identificativo univoco utilizzato per distinguere entità come utenti, auto e ordini.
- **UI/UX:** (User Interface/User Experience) Aspetti del design e dell'interazione utente.
- **SQL:** (Structured Query Language) Linguaggio per gestire dati in database relazionali.

Servizi e Funzionalità

- **Sottosistema di Autenticazione e Sicurezza:** Gestisce login, registrazione e protezione dei dati.
- **Sottosistema di Gestione delle Auto:** Permette l'organizzazione e l'aggiornamento dei dati delle auto.
- **Sottosistema di Ricerca e Filtro:** Consente la ricerca e il filtraggio delle auto basandosi su parametri specifici.
- **Sottosistema di Gestione Ordini:** Amministra gli ordini effettuati dai clienti.
- **Sottosistema di Interfaccia Utente (UI/UX):** Fornisce un'interfaccia intuitiva e responsiva.
- **Sottosistema di Gestione del Database:** Gestisce archiviazione e recupero dei dati persistenti.

Termini Tecnici

- **Caching:** Tecnica per migliorare le prestazioni memorizzando temporaneamente dati frequentemente utilizzati.
- **Autenticazione a Due Fattori:** Metodo opzionale per aumentare la sicurezza durante l'accesso alla piattaforma.