



Laurea Triennale in Informatica

Università di Salerno

Corso di Ingegneria del Software

Prof. Andrea De Lucia



Progetto EasyDrive

System Design Document (SDDD)

Versione 0.1

Data: 24/11/2025

Coordinatore del progetto:

Nome	Matricola
Mario Mascheri	0512120157
Claudio Brizio	0512119716

Partecipanti:

Nome	Matricola
Brizio Claudio	0512119716
Cannella Vincenzo	0512119065
Coscia Matteo	0512121210
Mascheri Mario	0512120157

Scritto da: Vincenzo Cannella

Cronologia delle Revisioni

Data	Versione	Descrizione	Autore
25/11/2025	0.1	Prima stesura e bozza del SDD	Claudio Brizio
09/12/2025	0.2	Seconda stesura del SDD	Claudio Brizio

Indice

1	Introduction	4
1.1	Purpose of the system	4
1.2	Design goals	4
1.2.1	Usabilità	4
1.2.2	Affidabilità (Reliability)	4
1.2.3	Prestazioni (Performance)	4
1.3	Portabilità	5
1.3.1	Supportabilità	5
1.4	Sicurezza:	5
1.5	Robustezza	5
1.6	Leggibilità	5
1.7	Definitions, acronyms, and abbreviations	5
1.8	References	6
1.9	Overview	6
2	Current software architecture	7
3	Proposed software architecture	7
3.1	Overview	7
3.2	Diagramma della decomposizione del sistema in sottosistemi	8
3.3	Subsystem decomposition	8
3.4	Hardware/software mapping	8
3.5	Deployment Diagram	9
3.6	Persistent data management	10
3.7	Access control and security	10
4	Gestione della Sicurezza	10
4.1	Matrice degli Accessi del Sistema EasyDrive	11
4.2	Global software control	11
4.3	Boundary conditions	11
5	Subsystem Services	12
5.1	Servizi del Sottosistema Catalogo Veicoli	12

5.2	Servizi del Sottosistema Gestione Utenti	12
5.3	Servizi del Sottosistema Prenotazioni e Noleggi	12
5.4	Servizi del Sottosistema Vendite	13
5.5	Servizi del Sottosistema Pagamenti	13
5.6	Servizi del Sottosistema Gestione Catalogo (Responsabili)	13
Glossary		13

1 Introduction

1.1 Purpose of the system

Lo scopo del sistema è progettare e realizzare una soluzione software integrata per la gestione delle attività di vendita e noleggio di automobili, rivolta a **concessionarie e autosaloni**. L'obiettivo principale è semplificare e digitalizzare le operazioni quotidiane, riducendo gli interventi manuali e ottimizzando l'efficienza operativa. Al tempo stesso, il sistema offrirà agli utenti finali un'interfaccia intuitiva e accessibile che consenta loro di consultare il catalogo veicoli, effettuare prenotazioni per il noleggio e completare l'acquisto di automobili in modo autonomo, sicuro e trasparente, migliorando l'esperienza sia della **concessionaria** sia dei clienti.

1.2 Design goals

1.2.1 Usabilità

Il sistema è progettato per essere semplice e intuitivo, in modo che gli utenti possano svolgere facilmente le operazioni principali: ricerca di veicoli, prenotazione, avvio del noleggio e acquisto. Per raggiungere questo obiettivo l'interfaccia adotta diverse soluzioni pratiche.

I menu e i pulsanti sono etichettati con parole chiare e comprensibili, così l'utente capisce subito quale azione eseguire. Nei form di inserimento dati — per esempio email, password o informazioni di pagamento — vengono mostrati messaggi immediati e comprensibili quando qualcosa non va o manca un'informazione, in modo da guidare l'utente a correggere l'errore.

Le operazioni più complesse sono presentate come procedure a passi: ad esempio, per acquistare o prenotare un'auto l'utente segue una sequenza guidata (es. "Step 1: seleziona veicolo", "Step 2: inserisci dati cliente", "Step 3: conferma pagamento"). Dopo ogni passaggio importante il sistema fornisce un riscontro visivo o testuale di conferma, così l'utente è certo che l'azione è stata completata correttamente.

Infine, nei campi da compilare vengono usati esempi e suggerimenti (placeholder) che mostrano il formato corretto dei dati, facilitando l'inserimento. Nel complesso, queste scelte rendono l'esperienza chiara, rapida e sicura per qualsiasi utente, anche senza formazione preventiva.

1.2.2 Affidabilità (Reliability)

Il sistema deve essere disponibile almeno per il 99% del tempo, con meccanismi di backup automatico giornaliero per prevenire la perdita dei dati. In caso di errore o malfunzionamento, deve essere in grado di ripristinare l'operatività entro 10 minuti.

1.2.3 Prestazioni (Performance)

Il sistema deve rispondere alle richieste degli utenti in meno di 2 secondi, assicurando un'esperienza fluida e senza ritardi anche in caso di un elevato numero di accessi simultanei.

1.3 Portabilità

Il sistema deve poter essere eseguito su piattaforme differenti (Windows, Linux, macOS) e supportare i principali browser web (Chrome, Edge, Firefox, Safari).

1.3.1 Supportabilità

Il sistema deve garantire un'elevata **scalabilità** e la capacità di adattarsi facilmente all'evoluzione del dominio applicativo. Deve essere progettato in modo da consentire una **integrazione semplice** con altri sistemi e da poter accogliere senza difficoltà nuove tecnologie. L'architettura del software dovrà inoltre essere **fortemente modulare**, così da permettere modifiche alla logica di business o alle componenti di sistema senza impattare in modo significativo sulle altre parti dell'applicazione.

1.4 Sicurezza:

Il sistema deve garantire la protezione dei dati personali tramite HTTPS, autenticazione sicura e cifratura di password e dati sensibili..

1.5 Robustezza

Il sistema deve essere **robusto**, ossia in grado di gestire correttamente input errati o non validi da parte degli utenti. In caso di dati non conformi, il sistema non deve accettare l'input, ma deve notificare l'utente tramite un messaggio chiaro, invitandolo a correggere l'errore. Questo comportamento garantisce la continuità del funzionamento del sistema senza compromettere l'integrità dei dati o la stabilità dell'applicazione.

1.6 Leggibilità

Il codice prodotto deve essere facilmente comprensibile. Ogni metodo e ogni campo non banale devono essere documentati in modo chiaro e appropriato, in modo da facilitarne la lettura e la comprensione da parte degli sviluppatori. Una buona leggibilità del codice supporta la manutenzione, la modifica e l'evoluzione del sistema nel tempo.

1.7 Definitions, acronyms, and abbreviations

RAD: Requirements Analysis Document

RNF: Requisito Non Funzionale

RF: Requisito Funzionale

UC: Caso d'Uso

SC: Scenario

MU: Mock-Up

ADM: Amministratore

CLI: Cliente

VEN: Venditore

CAR: Automobile

GUE: Guest

SDD: System Design Document

DBMS : DataBase Management System

1.8 References

- **Problem Statement – EasyDrive:**
Problem Statement (PDF)
- **Requirements Analysis Document (RAD) – EasyDrive:**
Requirements Analysis Document – RAD (PDF)

1.9 Overview

Il presente documento è strutturato in quattro sezioni principali volte a descrivere e analizzare il design del sistema proposto.

1. Introduzione

Questa sezione fornisce un contesto iniziale del progetto. Include la descrizione dello scopo del sistema, l'elenco e dei Design Goals (obiettivi di progettazione), le definizioni, gli acronimi e i riferimenti ad altri documenti.

2. Architettura del Sistema Corrente

Se applicabile, questa sezione analizza l'architettura del sistema esistente che si intende sostituire o migliorare. In assenza di un sistema precedente, fornisce una panoramica delle architetture correnti in sistemi simili.

3. Architettura del Sistema Proposto

Questa è la sezione principale del documento e descrive l'architettura del sistema da sviluppare. È suddivisa in modo da affrontare tutte le decisioni chiave del design:

Decomposizione dei Sottosistemi: Descrive la suddivisione del sistema in diversi sottosistemi.

Mapping Hardware/Software: In questa sezione vengono definite le piattaforme hardware su cui il sistema dovrà operare.

Una volta stabilite le piattaforme fisiche, si procede al mapping dei sottosistemi software su tali risorse.

Gestione Dati Persistenti:

La sezione Gestione Dati Persistenti ha il compito di descrivere i dati persistenti che verranno conservati dal sistema e l'infrastruttura di gestione necessaria per la loro archiviazione e manipolazione.

Controllo degli Accessi e Sicurezza;

Questa sezione descrive il modello degli utenti del sistema in termini di una matrice degli accessi (access matrix). La matrice definisce i diritti di accesso (Access Right) e modella le autorizzazioni per le classi del sistema. Vengono

specificate le informazioni a cui i singoli attori possono accedere e quali sono le operazioni che questi hanno la possibilità di effettuare. Vengono inoltre trattate le politiche di sicurezza, inclusa la selezione del meccanismo di autenticazione e l'eventuale uso della crittografia.

Controllo Globale del Software (Global Software Control):

Questa sezione descrive come è implementato il controllo globale del software.

Condizioni Boundary (Boundary Conditions) Questo paragrafo descrive le condizioni limite e il comportamento non stazionario del sistema. In particolare, vengono specificate le procedure di start-up (inizializzazione) e shutdown (terminazione) del sistema, oltre alla gestione dei fallimenti critici (error behavior).

4. Servizi dei Sottosistemi Questa sezione elenca i servizi forniti da ciascun sottosistema in termini di operazioni.

Glossario Fornisce definizioni esplicite per tutti i termini tecnici, gli acronimi e le abbreviazioni utilizzati, supportando la leggibilità e riducendo le incomprensioni

2 Current software architecture

Poiché EasyDrive non sostituisce un sistema precedente, questa sezione presenta una breve analisi delle architetture utilizzate dai software più diffusi nel settore del noleggio e della gestione automobili

Software simili e architetture comuni

Molti gestionali per il noleggio auto attualmente sul mercato (come Renthub, SimplyRent, AgileRent, Camasys o RentSys) adottano architetture molto simili tra loro. In generale:

Sono applicazioni web: si usano tramite browser, senza installazioni locali.

Funzionano in cloud: il sistema gira su un web server remoto che gestisce tutta la logica applicativa.

Utilizzano un database centralizzato: i dati della flotta, delle prenotazioni e dei clienti sono conservati in un DB server.

Sono modulari: di solito hanno moduli separati per prenotazioni, flotta, contratti, fatturazione e reportistica.

Offrono API o plugin: permettono l'integrazione con un sito web esterno per le prenotazioni online.

3 Proposed software architecture

3.1 Overview

Il nostro sistema è sviluppato secondo un'architettura client-server. In un'architettura client-server, uno o più server forniscono servizi a istanze di sottosistemi chiamati client. L'interazione avviene in modo che il client invia una richiesta al server, il quale esegue il servizio richiesto e restituisce il risultato.

Il client deve conoscere l'interfaccia del server, ovvero i servizi che il server può fornire, mentre il server non ha bisogno di conoscere l'interfaccia del client. In generale, la risposta del server viene fornita immediatamente, garantendo una comunicazione rapida ed efficiente.

Gli utenti interagiscono esclusivamente con il client, che funge da intermediario tra l'utente stesso e i servizi offerti dal server, senza accedere direttamente a quest'ultimo. Questo modello permette di separare chiaramente le responsabilità tra presentazione, logica applicativa e gestione dei dati, migliorando la modularità e la manutenibilità del sistema.

3.2 Diagramma della decomposizione del sistema in sottosistemi

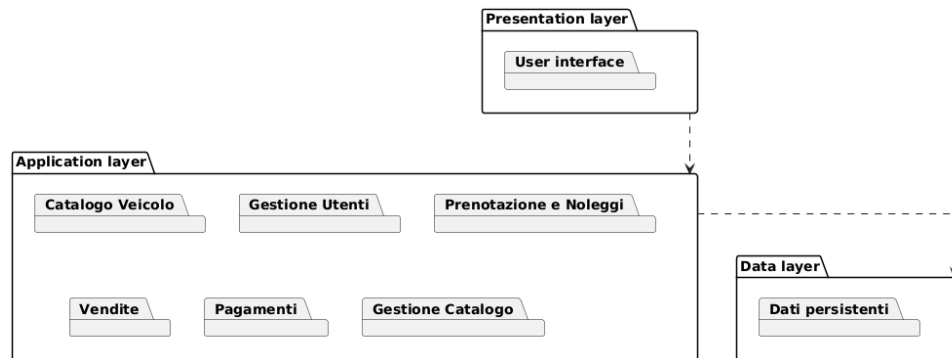


Figura 1: Diagramma della decomposizione del sistema in sottosistemi

3.3 Subsystem decomposition

- **Sottosistema Catalogo Veicoli:** gestisce le funzionalità di consultazione del catalogo, ricerca, filtraggio e visualizzazione dei dettagli dei veicoli, come definito nei casi d'uso dedicati alla navigazione del sistema.
- **Sottosistema Gestione Utenti:** si occupa delle operazioni di registrazione, autenticazione, gestione del profilo e gestione delle sessioni utente.
- **Sottosistema Prenotazioni e Noleggi:** gestisce il flusso di prenotazione dei veicoli a noleggio, incluse la conferma, la gestione delle disponibilità e l'aggiornamento dello stato del veicolo.
- **Sottosistema Vendite:** gestisce l'acquisto dei veicoli da parte degli utenti, comprendendo la conferma della vendita e la modifica dello stato del veicolo in base alle regole descritte nei casi d'uso.
- **Sottosistema Pagamenti:** gestisce l'intero processo di pagamento.
- **Sottosistema Gestione Catalogo (Responsabili):** fornisce ai responsabili le funzionalità per inserire, modificare e aggiornare i veicoli presenti nel catalogo.

3.4 Hardware/software mapping

Il sistema che si desidera sviluppare utilizzerà una struttura hardware composta da un Server che risponderà ai servizi richiesti dal Client.

Il Client è una macchina attraverso la quale un utente può collegarsi, usando una connessione a internet, per accedere al sistema. Le componenti hardware e software di cui ha bisogno il client sono un computer, un tablet oppure un qualsiasi dispositivo mobile dotato di connessione internet e di un browser web moderno in grado di interpretare HTML, CSS e JavaScript per la visualizzazione dell'interfaccia utente.

Per quanto riguarda il Server, vi è la necessità di avere a disposizione una macchina avente connessione a internet e con capacità di immagazzinare grandi quantità di dati. La macchina server gestirà la logica dell'applicazione (tramite PHP) e i dati persistenti inseriti nel database.

Per l'interazione e la gestione del servizio, sono necessarie le seguenti componenti software:

Un Web Server (come Apache o Nginx) per consentire la comunicazione con più client e instradare le richieste HTTP.

Il Runtime di PHP (PHP Interpreter), che esegue gli script contenenti la logica di business e genera la risposta dinamica da inviare al client.

Un DBMS, nello specifico MySQL, per consentire il salvataggio dei dati in maniera persistente.

Client e Server saranno connessi tramite il protocollo HTTP/HTTPS, con cui il client inoltra richieste al server e quest'ultimo provvederà a fornire i servizi richiesti.

3.5 Deployment Diagram

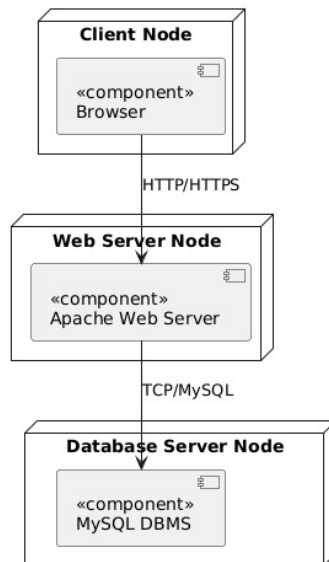


Figura 2: Deployment Diagram

3.6 Persistent data management

Per la memorizzazione dei dati di sistema è stato adottato un Database Management System (DBMS) Relazionale (MySQL). Questa scelta è motivata dalla necessità di garantire brevi tempi di risposta (basati su query ottimizzate) e di garantire un'elevata efficienza nell'archiviazione dei dati. Il DBMS include meccanismi per gestire l'archiviazione persistente e fornire un meccanismo di backup periodico. Ciò permette il ripristino dello stato del database a fronte di guasti software o hardware. Inoltre essendo un sistema multi-utente, il database è configurato per supportare l'accesso concorrente. La logica di accesso implementa specifiche politiche di sicurezza, che determinano i livelli di privilegio e le operazioni consentite per i diversi attori del sistema, proteggendo così la privacy e l'integrità dei dati.

3.7 Access control and security

4 Gestione della Sicurezza

La sicurezza del sistema EasyDrive è garantita attraverso una combinazione di meccanismi di autenticazione, controllo degli accessi e protezione dei dati sensibili. In dettaglio:

- **Autenticazione degli utenti:** Ogni utente deve autenticarsi tramite credenziali uniche (username e password). Le password sono memorizzate in forma crittografata nel database utilizzando algoritmi sicuri di hashing (ad esempio SHA-256).
- **Controllo degli accessi:** Il sistema implementa una matrice di accesso basata sui ruoli degli utenti. Ogni ruolo (ad esempio amministratore, cliente, operatore) ha permessi specifici che determinano quali funzionalità e dati può leggere, modificare o cancellare.
- **Protezione dei dati in transito:** Tutte le comunicazioni tra client e server avvengono tramite connessione sicura HTTPS, garantendo la crittografia dei dati durante il trasferimento.
- **Protezione dei dati a riposo:** I dati sensibili, come informazioni personali degli utenti o dettagli di pagamento, sono memorizzati crittografati nel database, in modo da ridurre il rischio in caso di accesso non autorizzato.

Queste misure assicurano che EasyDrive possa garantire la riservatezza, l'integrità e la disponibilità dei dati, proteggendo le informazioni degli utenti e prevenendo accessi non autorizzati.

4.1 Matrice degli Accessi del Sistema EasyDrive

La Tabella seguente mostra la matrice degli accessi del sistema *EasyDrive*, in cui per ogni ruolo vengono indicati i permessi associati alle risorse del sistema.

Attore	Account	Catalogo Veicoli	Prenotazioni	Vendite	Gestione Catalogo
Guest	<ul style="list-style-type: none">• Registrazione	<ul style="list-style-type: none">• Visualizzare veicoli• Visualizzare dettagli• Ricerca veicoli	–	–	–
Cliente	<ul style="list-style-type: none">• Login• Logout• Gestione profilo	<ul style="list-style-type: none">• Visualizzare veicoli• Visualizzare dettagli• Ricerca veicoli	<ul style="list-style-type: none">• Creare prenotazione• Visualizzare stato• Confermare prenotazione	<ul style="list-style-type: none">• Avvio processo di vendita• Conferma vendita	–
Responsabile Catalogo	<ul style="list-style-type: none">• Login• Logout	<ul style="list-style-type: none">• Visualizzare veicoli• Visualizzare dettagli• Ricerca veicoli• Aggiornare disponibilità post-vendita	–	–	<ul style="list-style-type: none">• Inserire veicolo• Modificare veicolo• Rimuovere veicolo• Aggiornare disponibilità
Responsabile Vendite / Gestore Ordini	<ul style="list-style-type: none">• Login• Logout	–	–	<ul style="list-style-type: none">• Inserimento ordine di vendita• Visualizzare tutti gli ordini• Confermare ordine• Modificare ordine	–

Tabella 1: Matrice degli Accessi del sistema EasyDrive

4.2 Global software control

Il sistema EasyDrive adotta un modello di Controllo Centralizzato, nel quale è il Server a gestire il flusso di lavoro per ogni azione eseguita dall'utente. Proprio per questa struttura, il controllo software globale risulta essere di tipo procedure-driven. Questo significa che ogni funzionalità del sistema viene attivata e gestita attraverso una sequenza di istruzioni ben definita, stabilita dal codice lato server.

Questo approccio deriva direttamente dal modello di esecuzione di PHP nel suo utilizzo tradizionale. PHP opera seguendo il paradigma request-response, tipico dei sistemi web basati su HTTP: quando l'utente compie un'azione sull'interfaccia (ad esempio clicca un pulsante o invia un form), il browser invia una richiesta al server. A questo punto il server — tramite il webserver e il motore PHP — esegue lo script associato a quella specifica richiesta. Lo script viene processato dall'inizio alla fine in modo sequenziale, seguendo esattamente l'ordine delle istruzioni stabilite dal programmatore. Terminata l'elaborazione, il server genera una risposta tipicamente una pagina HTML e conclude l'esecuzione.

4.3 Boundary conditions

Data l'architettura di sistema Client/Server, l'applicazione EasyDrive viene eseguita all'interno di un ambiente web predefinito (composto da Web Server, interprete PHP e servizi di rete associati).

Di conseguenza, tutte le operazioni relative all'avvio, alla terminazione (shutdown) e alla configurazione dell'infrastruttura sono intrinsecamente gestite dal server stesso. È il Web Server ad assumere la responsabilità del ciclo di vita dell'applicazione: esso carica l'interprete PHP in risposta a una Richiesta HTTP in ingresso e provvede alla disallocazione delle risorse al completamento dell'elaborazione.

Di conseguenza, EasyDrive non ha bisogno di funzioni interne che gestiscano il server, né di attori dedicati a questo compito. Tutte le attività come configurazione, manutenzione e controllo del server vengono gestite automaticamente dall'ambiente in cui l'applicazione è eseguita. Per questo motivo non esistono casi d'uso che descrivono queste operazioni: non sono azioni svolte dagli utenti sul sistema e non appartengono alla logica dell'applicazione.

5 Subsystem Services

Questa sezione descrive i servizi forniti da ciascun sottosistema. I servizi rappresentano le operazioni esposte da un sottosistema agli altri sottosistemi, costituendo l'interfaccia funzionale interna del sistema.

5.1 Servizi del Sottosistema Catalogo Veicoli

- **Fornitura elenco veicoli** — servizio che restituisce l'elenco dei veicoli disponibili secondo eventuali criteri di filtro.
- **Fornitura dettagli veicolo** — permette ad altri sottosistemi di ottenere le informazioni complete di un veicolo.
- **Ricerca veicoli** — consente la ricerca nel catalogo sulla base di criteri specifici.
- **Verifica disponibilità veicolo** — mette a disposizione lo stato di disponibilità di un veicolo.

5.2 Servizi del Sottosistema Gestione Utenti

- **Autenticazione utente** — servizio per la verifica dell'identità dell'utente.
- **Registrazione utente** — permette la creazione di un nuovo profilo utente.
- **Fornitura dati utente** — rende disponibili le informazioni associate a un utente registrato.
- **Aggiornamento dati utente** — servizio per modificare le informazioni del profilo esistente.

5.3 Servizi del Sottosistema Prenotazioni e Noleggi

- **Creazione prenotazione** — servizio che genera una richiesta di prenotazione per un veicolo.
- **Conferma prenotazione** — utilizzato dopo l'esito positivo di un pagamento.
- **Aggiornamento disponibilità per noleggio** — aggiorna lo stato del veicolo dopo una prenotazione.
- **Fornitura stato prenotazione** — permette la consultazione dello stato corrente di una prenotazione.

5.4 Servizi del Sottosistema Vendite

- **Avvio processo di vendita** — servizio che consente l’inizio della procedura di acquisto di un veicolo.
- **Conferma vendita** — servizio eseguito a seguito di un pagamento completato con successo.
- **Aggiornamento disponibilità post-vendita** — modifica lo stato del veicolo dopo la vendita.

5.5 Servizi del Sottosistema Pagamenti

- **Elaborazione pagamento** — servizio per l’invio dei dati di pagamento al sistema esterno.
- **Gestione esito pagamento** — fornisce l’esito della transazione (successo, fallimento, timeout).
- **Verifica preliminare metodi di pagamento** — controlla la validità dei dati prima dell’invio al servizio esterno.

5.6 Servizi del Sottosistema Gestione Catalogo (Responsabili)

- **Inserimento veicolo** — permette di aggiungere un nuovo veicolo al catalogo.
- **Modifica informazioni veicolo** — consente l’aggiornamento dei dati già registrati.
- **Aggiornamento disponibilità** — servizio di modifica dello stato del veicolo.
- **Rimozione veicolo** — consente di eliminare un veicolo dal catalogo.

Glossary

Nella presente sezione sono raccolti i termini e le sigle utilizzati nel documento che necessitano di una definizione.

Veicolo Automobile destinata alla vendita o al noleggio. Ogni veicolo è identificato da un codice univoco e da informazioni tecniche come modello, marca, anno di immatricolazione e stato di disponibilità.

Catalogo Elenco dei veicoli disponibili all’interno dell’agenzia, consultabile sia dai clienti che dai dipendenti tramite l’interfaccia del sistema.

Prenotazione Operazione che consente a un cliente di riservare temporaneamente un veicolo in attesa di conferma, pagamento o firma del contratto.

Contratto Documento digitale che formalizza un’operazione di vendita o noleggio tra l’agenzia e il cliente. Contiene dati relativi al veicolo, alle condizioni economiche e alla durata del servizio.

Cliente Utente esterno che accede alla piattaforma per consultare il catalogo dei veicoli, effettuare prenotazioni, acquisti o noleggi.

Utente Registrato Cliente che possiede un account sul sistema EasyDrive e può accedere alle funzionalità riservate come la gestione delle proprie prenotazioni e contratti.

Amministratore Utente con privilegi avanzati, responsabile della configurazione generale del sistema, della gestione dei dipendenti e del controllo delle attività dell’agenzia.

Piattaforma Applicazione web EasyDrive che integra tutte le funzionalità necessarie alla gestione digitale delle attività di un'agenzia automobilistica.

Dashboard Interfaccia principale riservata ai dipendenti e agli amministratori, che consente di visualizzare e gestire in tempo reale le informazioni relative a veicoli, prenotazioni e contratti.

Pagamento Operazione che consente al cliente di completare una transazione di vendita o noleggio in modo sicuro attraverso la piattaforma.

Account Rappresentazione digitale di un utente nel sistema EasyDrive, contenente dati personali, credenziali di accesso e storico delle operazioni effettuate.

Mock-up Rappresentazione visiva dell'interfaccia utente del sistema, utile per mostrare al committente l'aspetto grafico e il modo in cui l'applicazione sarà utilizzata dall'utente finale.