



Laurea Triennale in Informatica

Università di Salerno

Corso di Ingegneria del Software

Prof. Andrea De Lucia



Progetto EasyDrive System Design Document (SDD)

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
28/01/2026	0.3	Stesura finale 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)	5
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 Pattern MVC	8
3.3 Diagramma della decomposizione del sistema in sottosistemi	8
3.4 Subsystem Decomposition	8
3.5 Hardware/software mapping	9
3.6 Deployment Diagram	10
3.7 Persistent data management	11
3.8 Access control and security	11
4 Gestione della Sicurezza	11
4.1 Matrice degli Accessi del Sistema EasyDrive	11
4.2 Global software control	12
4.3 Boundary conditions	13
5 Subsystem Services	13

5.1	Servizi del Sottosistema di Account	13
5.2	Servizi del Sottosistema Catalogo Veicoli	13
5.3	Servizi del Sottosistema Vendite	14
5.4	Servizi del Sottosistema Prenotazioni e Noleggi	14
5.5	Servizi del Sottosistema Pagamenti	14
6	Glossario	15

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 offrire un'esperienza d'uso immediata, consentendo agli utenti di completare le operazioni principali — ricerca, prenotazione, noleggio e acquisto — con la massima fluidità. Per garantire questo standard, l'interfaccia adotta le seguenti soluzioni strategiche:

1. Navigazione Intuitiva e Linguaggio Chiaro L'architettura delle informazioni si basa su menu e pulsanti etichettati in modo esplicito. L'uso di una terminologia comune e priva di tecnicismi permette all'utente di identificare istantaneamente l'azione da compiere, riducendo i tempi di apprendimento.
2. Validazione dei Dati e Feedback in Tempo Reale Per prevenire errori durante la compilazione dei form (come email, password o dati di pagamento), il sistema integra meccanismi di validazione istantanea:

Messaggi di errore chiari: Segnalano immediatamente l'informazione mancante o errata.

Suggerimenti contestuali: Guidano l'utente verso la correzione, evitando la frustrazione di dover ricominciare la procedura.

Placeholder esplicativi: I campi di inserimento mostrano esempi del formato corretto (es. nome@esempio.it), semplificando ulteriormente l'interazione.

3. Workflow Guidati per Operazioni Complesse Le procedure più articolate, come la prenotazione o l'acquisto di un veicolo, sono suddivise in processi a passi sequenziali (Step-by-Step). Questo approccio:

Riduce il carico cognitivo dell'utente.

Fornisce un percorso strutturato (es. Selezione Veicolo > Dati Cliente > Pagamento).

Garantisce certezza sull'avanzamento grazie a riscontri visivi e testuali inviati al termine di ogni passaggio chiave.

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 RentSyst) 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 Pattern MVC

Il sistema sarà costruito seguendo il pattern architettonico **MVC (Model-View-Controller)**. In questo pattern le componenti del sistema appartengono ai tre tipi principali:

- **Model:** rappresenta le informazioni del dominio applicativo e permette di accedervi.
- **View:** si occupa di presentare all’utente i dati contenuti nei Model, fornendo l’interfaccia grafica o testuale.
- **Controller:** riceve i comandi dell’utente dalla View e li applica, modificando lo stato dei Model e aggiornando le View di conseguenza.

3.3 Diagramma della decomposizione del sistema in sottosistemi

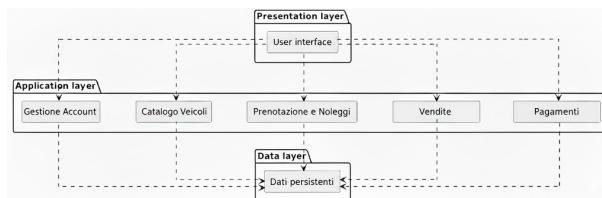


Figura 1: Diagramma della decomposizione del sistema in sottosistemi

Per semplificare lo sviluppo del sistema, l’architettura è stata organizzata in tre strati principali:

Presentation Layer: si occupa delle interfacce grafiche con le quali l’utente interagisce, raccogliendo input e visualizzando i risultati delle operazioni.

Application Layer: gestisce tutte le richieste inviate dall’utente tramite il Presentation Layer, elaborando i dati e restituendo i risultati delle operazioni richieste.

Data Layer: si occupa della gestione e della persistenza dei dati del sistema, garantendo che le informazioni siano accessibili e aggiornate in modo sicuro.

3.4 Subsystem Decomposition

Il sistema **EasyDrive** è strutturato nei seguenti sottosistemi, progettati per massimizzare la coesione e ridurre l’accoppiamento tra le diverse aree funzionali, in accordo con i principi di progettazione orientata agli oggetti e con i requisiti definiti nel RAD.

- **Sottosistema Gestione Account** — gestisce il ciclo di vita dell’identità digitale degli utenti, includendo le funzionalità di registrazione, autenticazione, logout. Questo sottosistema è responsabile esclusivamente degli aspetti di sicurezza e identità e non include logica di business relativa a vendite o noleggi.
- **Sottosistema Catalogo Veicoli** — centralizza la gestione delle informazioni relative ai veicoli presenti nel sistema. Fornisce funzionalità di consultazione, ricerca e filtraggio dei veicoli disponibili, nonché servizi per la gestione e l’aggiornamento dei dati dei veicoli da parte degli utenti autorizzati. Il sottosistema non gestisce operazioni di acquisto, noleggio o pagamento.
- **Sottosistema Prenotazioni e Noleggi** — gestisce il processo di noleggio dei veicoli, comprendendo la creazione delle prenotazioni, la definizione del periodo di utilizzo, la conferma o l’annullamento delle operazioni e la gestione dello stato del noleggio. La prenotazione rappresenta uno stato iniziale del ciclo di vita del noleggio ed è quindi modellata all’interno dello stesso sottosistema.
- **Sottosistema Vendite** — gestisce il processo di acquisto definitivo dei veicoli da parte degli utenti. Include la creazione e la gestione degli ordini di vendita, la visualizzazione dello storico delle transazioni e l’aggiornamento dello stato degli ordini (CREATO, PAGATO, ANNULLATO). Il sottosistema è distinto da quello dei Noleggi al fine di preservare la chiarezza semantica e la separazione delle responsabilità di dominio.
- **Sottosistema Pagamenti** — gestisce il processo di pagamento in modo indipendente dagli altri sottosistemi. Fornisce servizi per la validazione dei dati di pagamento, l’elaborazione interna delle transazioni e la comunicazione dell’esito del pagamento ai sottosistemi di Vendite e Noleggi, mantenendo isolata la logica economica dal resto del sistema.

3.5 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.

3.6 Deployment Diagram

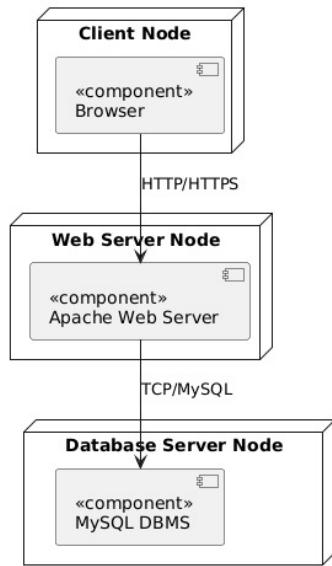


Figura 2: Deployment Diagram

3.7 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.

3.8 Access control and security

Questa sezione definisce le strategie e i meccanismi adottati per proteggere le risorse del sistema e garantire che solo gli utenti autorizzati possano accedere a dati e funzionalità specifiche. Il design della sicurezza si articola in due pilastri principali: le politiche di sicurezza (autenticazione e protezione dati) e il modello di controllo degli accessi.

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.

Tabella 1: Matrice degli Accessi del Sistema EasyDrive - Revisione Finale

Attore	Account	Catalogo Veicoli	Vendite	Prenotazioni e Noleggi	Pagamenti
Guest	Registrazione utente	Fornitura elenco veicoli Fornitura dettagli veicolo Ricerca veicoli noleggiani	-	-	-
Cliente	Autenticazione utente Logout utente Fornitura dati utente autenticato	Fornitura elenco veicoli Fornitura dettagli veicolo Ricerca veicoli noleggiani	Creazione ordine di vendita Fornitura storico vendite	Creazione prenotazione di noleggio Calcolo preventivo di noleggio Annullamento ordine Fornitura dettagli noleggio	Inserimento dati di pagamento
Gestore Veicoli	Autenticazione utente Logout utente	Inserimento veicolo Rimozione veicolo Aggiornamento disponibilità veicolo Modifica dati veicolo Fornitura elenco veicoli Fornitura dettagli veicolo	-	-	-
Gestore Vendite	Autenticazione utente Logout utente	Fornitura elenco veicoli Fornitura dettagli veicolo	Conferma vendita Aggiornamento stato ordine di vendita Fornitura storico vendite	Conferma noleggio Annullamento ordine Aggiornamento stato noleggio Fornitura dettagli noleggio	-

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 un 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 di Account

- **Autenticazione utente** — consente ad un Utente di accedere al sistema tramite l’inserimento di username e password validi.
- **Registrazione utente** — consente ad un Utente non registrato di creare un nuovo account fornendo le informazioni richieste dal sistema.
- **Logout utente** — consente ad un Utente autenticato di terminare la propria sessione e uscire dal sistema in modo sicuro.
- **Fornitura dati utente autenticato** consente di visualizzare i dati di un utente autenticato

5.2 Servizi del Sottosistema Catalogo Veicoli

- **Fornitura elenco veicoli** — consente ad un Utente di visualizzare l’elenco dei veicoli presenti nel catalogo secondo eventuali criteri di filtro.
- **Fornitura dettagli veicolo** — consente ad un Utente di visualizzare le informazioni complete relative a un veicolo specifico.
- **Ricerca veicoli noleggibili** — consente ad un Utente di effettuare ricerche nel catalogo sulla base di criteri specifici.
- **Inserimento veicolo** — consente ad un Utente con ruolo di GestoreVeicoli di inserire un nuovo veicolo nel catalogo.

- **Rimozione veicolo** — consente ad un Utente con ruolo di GestoreVeicoli di rimuovere un veicolo dal catalogo.
- **Aggiornamento disponibilità veicolo** — consente ad un utente con il ruolo di GestoreVeicoli di aggiornare la disponibilità del veicolo.
- **Modifica dati veicolo** — consente ad un utente con il ruolo di GestoreVeicoli di modificare i dati del veicolo.

5.3 Servizi del Sottosistema Vendite

- **Creazione ordine di vendita** — consente a un Utente di avviare il processo di acquisto di un veicolo.
- **Conferma vendita** — consente di confermare un ordine di vendita a seguito dell'esito positivo del pagamento.
- **Aggiornamento stato ordine di vendita** — aggiorna lo stato dell'ordine (CREATO, PAGATO, ANNUL-LATO).
- **Fornitura storico vendite** — consente a un Utente autenticato di visualizzare lo storico degli ordini di vendita effettuati.

5.4 Servizi del Sottosistema Prenotazioni e Noleggi

- **Creazione prenotazione di noleggio** — consente a un Utente di effettuare una prenotazione per il noleggio di un veicolo selezionando un periodo di utilizzo e specificando ulteriori parametri di ricerca.
- **Calcolo preventivo di noleggio** — calcola il costo del noleggio sulla base del periodo selezionato.
- **Conferma noleggio** — consente di confermare una prenotazione a seguito dell'esito positivo del pagamento.
- **Annnullamento ordine** — consente di annullare una prenotazione di noleggio.
- **Aggiornamento stato noleggio** — aggiorna lo stato del noleggio (PRENOTATO, CONFER-MATO, ANNULATO).
- **Fornitura dettagli noleggio** — consente di visualizzare i dettagli relativi a una prenotazione o a un noleggio.

5.5 Servizi del Sottosistema Pagamenti

- **Inserimento dati di pagamento** — Permette di inserire i dati della carta di credito, necessari per effettuare il checkout.
- **Verifica dati di pagamento** — verifica la correttezza e la completezza dei dati di pagamento forniti dall'utente.
- **Gestione esito pagamento** — fornisce l'esito della transazione (successo o fallimento).

6 Glossario

Nella presente sezione sono raccolti e definiti i termini, gli attori e i concetti chiave utilizzati nel documento, al fine di garantire una comprensione univoca delle funzionalità del sistema **EasyDrive**.

Account	Rappresentazione digitale di un utente nel sistema, contenente credenziali di accesso, dati personali e lo storico delle operazioni effettuate (noleggi e acquisti).
Amministratore	Utente con privilegi avanzati (staff dell'autosalone), responsabile della configurazione del sistema, della gestione del catalogo e del monitoraggio delle attività aziendali.
Catalogo	Elenco digitale dei veicoli disponibili, aggiornato in tempo reale e consultabile tramite filtri personalizzati come marca, modello, anno e prezzo.
Cliente	Utente esterno che interagisce con la piattaforma per consultare i veicoli, effettuare prenotazioni o completare acquisti e contratti di noleggio.
Contratto	Documento digitale che formalizza l'accordo di vendita o noleggio, includendo i dettagli del veicolo, le condizioni economiche e i termini di durata.
Dashboard	Interfaccia di controllo dedicata allo staff e agli amministratori per la gestione operativa e la visualizzazione rapida dei dati relativi a veicoli e transazioni.
Mock-up	Rappresentazione grafica statica delle interfacce utente, utilizzata per illustrare il design, il layout e l'esperienza d'uso dell'applicazione.
Pagamento	Processo di transazione finanziaria eseguito elettronicamente per finalizzare l'acquisto di un veicolo o il deposito per un noleggio.
Piattaforma	L'applicazione software EasyDrive nel suo insieme, comprensiva di interfaccia utente, logica di gestione dati e database.
Prenotazione	Operazione che permette di riservare un veicolo per un determinato periodo, impedendone la vendita o il noleggio ad altri utenti in attesa della firma del contratto.
Utente Registrato	Cliente che ha creato un profilo verificato sulla piattaforma, ottenendo l'accesso a funzionalità riservate come la gestione dei propri contratti attivi.
Veicolo	L'entità fisica (autovettura) censita nel sistema, dotata di specifiche tecniche, codice identificativo univoco e stato di disponibilità aggiornato.