**Software Requirements**

**Specification**

**For**

*CAR-SERVICE*

***Prepared by Francesca Esposito (A13002071)***

***Esame di Sistemi Web e Basi Dati***

***INTRODUZIONE***

In questo documento verrà data una panoramica su una web application di car-sharing per la gestione e manutenzione di un database di un’officina al fine di registrare i clienti, le auto, i fornitori, gli interventi e i pezzi di ricambio e verificare le disponibilità di pezzi.

Questa applicazione ha come scopo primo e ultimo quello di supportare il proprietario nelle operazioni quotidiane legate ai servizi di gestione dell’officina e fornisce una visione completa e dettagliata delle informazioni pertinenti per una buona gestione di essa.

**Specifica dei Requisiti per il Sistema di Gestione dell'Autofficina**

**1.** *Descrizione Generale*

Il sistema di gestione dell'autofficina è progettato per aiutare i proprietari delle officine a gestire facilmente i loro clienti, auto, interventi, fornitori e pezzi di ricambio. L'applicazione web fornirà funzionalità per l'autenticazione dell'utente, la registrazione e la visualizzazione dei dati inseriti, assicurando la protezione e l'integrità dei dati.

**2. Requisiti Funzionali**

**2.1 Gestione degli Utenti**

* **Registrazione Utente**
  + Il sistema deve permettere la registrazione di nuovi utenti (proprietari dell'officina).
  + Ogni utente deve essere associato al nome dell'officina.
* **Autenticazione Utente**
  + Il sistema deve permettere agli utenti di effettuare il login utilizzando nome utente e password.
  + Dopo il login, gli utenti devono essere reindirizzati alla pagina di inserimento dati.
* **Logout**
  + Il sistema deve permettere agli utenti di effettuare il logout e reindirizzarli alla homepage.

**2.2 Gestione dei Clienti**

* **Inserimento Clienti**
  + Il sistema deve permettere l'inserimento di nuovi clienti con nome, cognome, e-mail e telefono.
* **Visualizzazione Clienti**
  + Il sistema deve permettere la visualizzazione di tutti i clienti registrati.

**2.3 Gestione delle Auto**

* **Inserimento Auto**
  + Il sistema deve permettere l'inserimento di nuove auto con targa, modello, anno e ID cliente associato.
* **Visualizzazione Auto**
  + Il sistema deve permettere la visualizzazione di tutte le auto registrate.

**2.4 Gestione dei Fornitori**

* **Inserimento Fornitori**
  + Il sistema deve permettere l'inserimento di nuovi fornitori con nome, contatto ed e-mail.
* **Visualizzazione Fornitori**
  + Il sistema deve permettere la visualizzazione di tutti i fornitori registrati.

**2.5 Gestione dei Pezzi di Ricambio**

* **Inserimento Pezzi di Ricambio**
  + Il sistema deve permettere l'inserimento di nuovi pezzi di ricambio con nome, codice, prezzo e ID fornitore associato.
* **Visualizzazione Pezzi di Ricambio**
  + Il sistema deve permettere la visualizzazione di tutti i pezzi di ricambio registrati.

**2.6 Gestione degli Interventi**

* **Inserimento Interventi**
  + Il sistema deve permettere l'inserimento di nuovi interventi con descrizione, data, ID auto e ID pezzo di ricambio associato.
* **Visualizzazione Interventi**
  + Il sistema deve permettere la visualizzazione di tutti gli interventi registrati.

**3. Requisiti Non Funzionali**

* **Sicurezza**
  + Le password devono essere criptate utilizzando una funzione di hashing sicura.
  + L'accesso alle pagine protette deve essere consentito solo agli utenti autenticati.
* **Usabilità**
  + L'interfaccia utente deve essere intuitiva e facile da usare.
  + Le pagine devono essere responsive e accessibili da vari dispositivi (desktop, tablet, smartphone).
* **Affidabilità**
  + Il sistema deve garantire l'integrità dei dati con adeguati controlli di validità.
  + Il sistema deve gestire correttamente gli errori e fornire messaggi di errore utili agli utenti.
* **Manutenibilità**
  + Il codice deve essere ben strutturato e documentato per facilitare la manutenzione e l'aggiornamento futuro.
* **Performance**
  + Il sistema deve rispondere alle richieste degli utenti in modo tempestivo e garantire tempi di caricamento ridotti.

**4. Requisiti di Implementazione**

* **Tecnologie Utilizzate**
  + Backend: PHP
  + Database: MySQL
  + Frontend: HTML, CSS, JavaScript
  + Web Server: Apache (utilizzando XAMPP)
* **Configurazione del Database**
  + Creazione di tabelle per utenti, clienti, auto, fornitori, pezzi di ricambio e interventi.
  + Definizione delle relazioni tra le tabelle per garantire l'integrità referenziale.
* **Ambiente di Sviluppo**
  + L'ambiente di sviluppo deve includere un server locale (es. XAMPP) per testare l'applicazione.
  + Gli sviluppatori devono utilizzare un sistema di controllo versione (es. Git) per gestire il codice sorgente.

**Documento di Progettazione**

Questo documento fornisce la progettazione dettagliata del sistema di gestione per un’autofficina. Include i requisiti del sistema, la progettazione UML dei componenti software, e lo sviluppo della progettazione concettuale e logica del database.

Il sistema di gestione per un’autofficina permette ai proprietari di autofficina di gestire clienti, auto, interventi, pezzi di ricambio. Include funzionalità di registrazione, login/logout, inserimento e visualizzazione dati.

**Progettazione Concettuale del Database**

***Modello E-R (Entità-Relazione)***

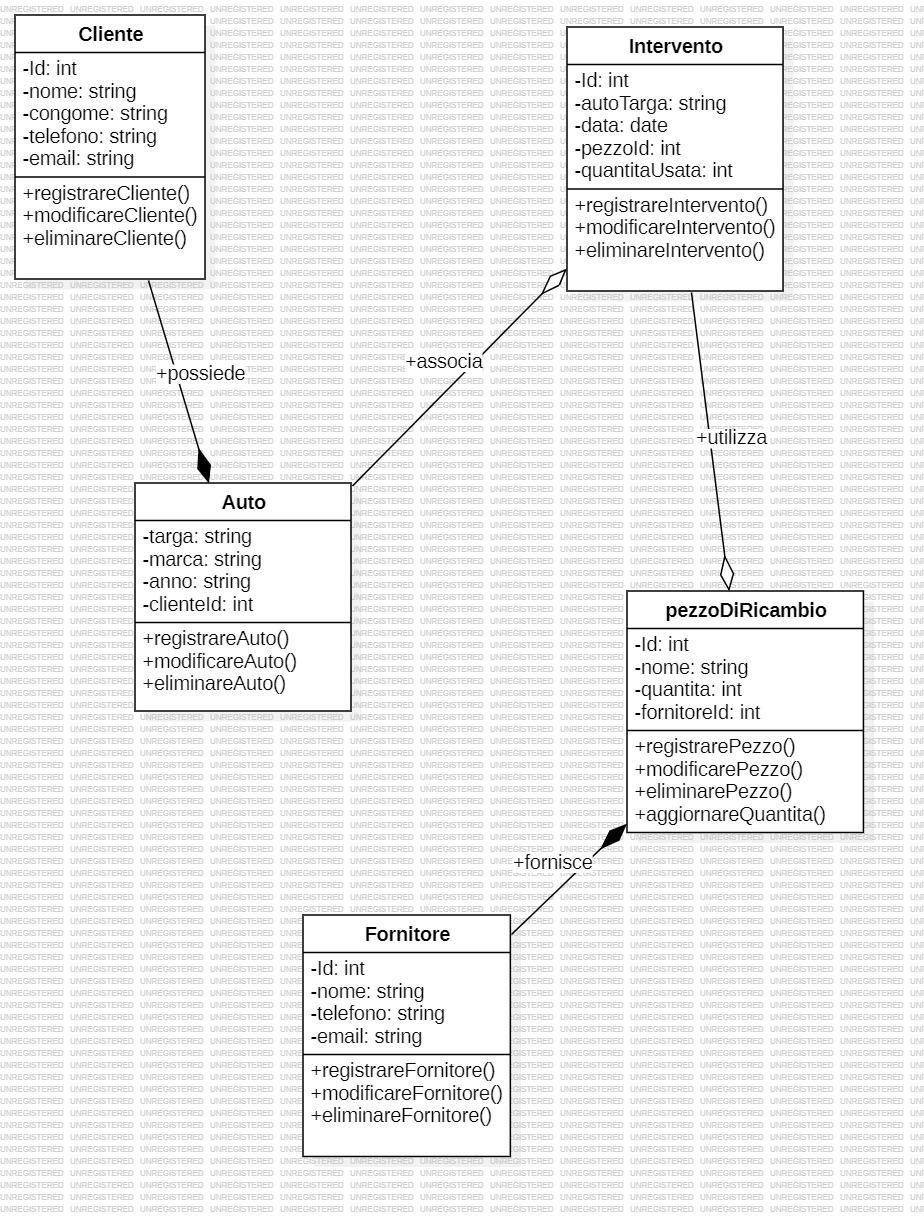
* PROPRIETARIO (user)
* **Attributi**: Id\_Cliente (PK), username, password, nome\_officina
* CLIENTE:
* **Attributi**: ID\_Cliente (PK), nome, cognome, e-mail, telefono
* AUTO:
* **Attributi**: ID\_Auto (PK), ID\_Cliente (PK), modello, anno, targa
* FORNITORE:
* **Attributi**: ID\_Fornitore (PK), nome, contatto, e-mail, telefono
* PEZZO DI RICAMBIO:
* **Attributi**: ID\_Pezzo (PK), nome, descrizione, prezzo, ID\_Fornitore (FK)
* INTERVENTI:
* **Attributi**: ID\_intervento (PK), ID\_Auto (FK), ID\_pezzo (FK), Descrizione, data

***Relazioni***

* Un **user** (proprietario) gestisce molte **auto**
* Un **cliente** possiede molte **auto**
* **Un’auto** può avere più **interventi**
* Un **intervento** può avere più **pezzi di ricambio**
* Un **fornitore** fornisce molti **pezzi di ricambio**

**Diagramma delle Classi**

Dal seguente diagramma delle classi si evincono le principali classi, con i relativi metodi e attributi oltre alla natura dei diversi tipi di legami che intercorrono fra le varie classi.



## ***Modello concettuale: diagramma ER***

## Il modello E-R (Entity-Relationship) è un approccio di modellazione dei dati che descrive la struttura logica dei database. Utilizza una combinazione di entità, attributi e relazioni per rappresentare i dati e le loro connessioni.

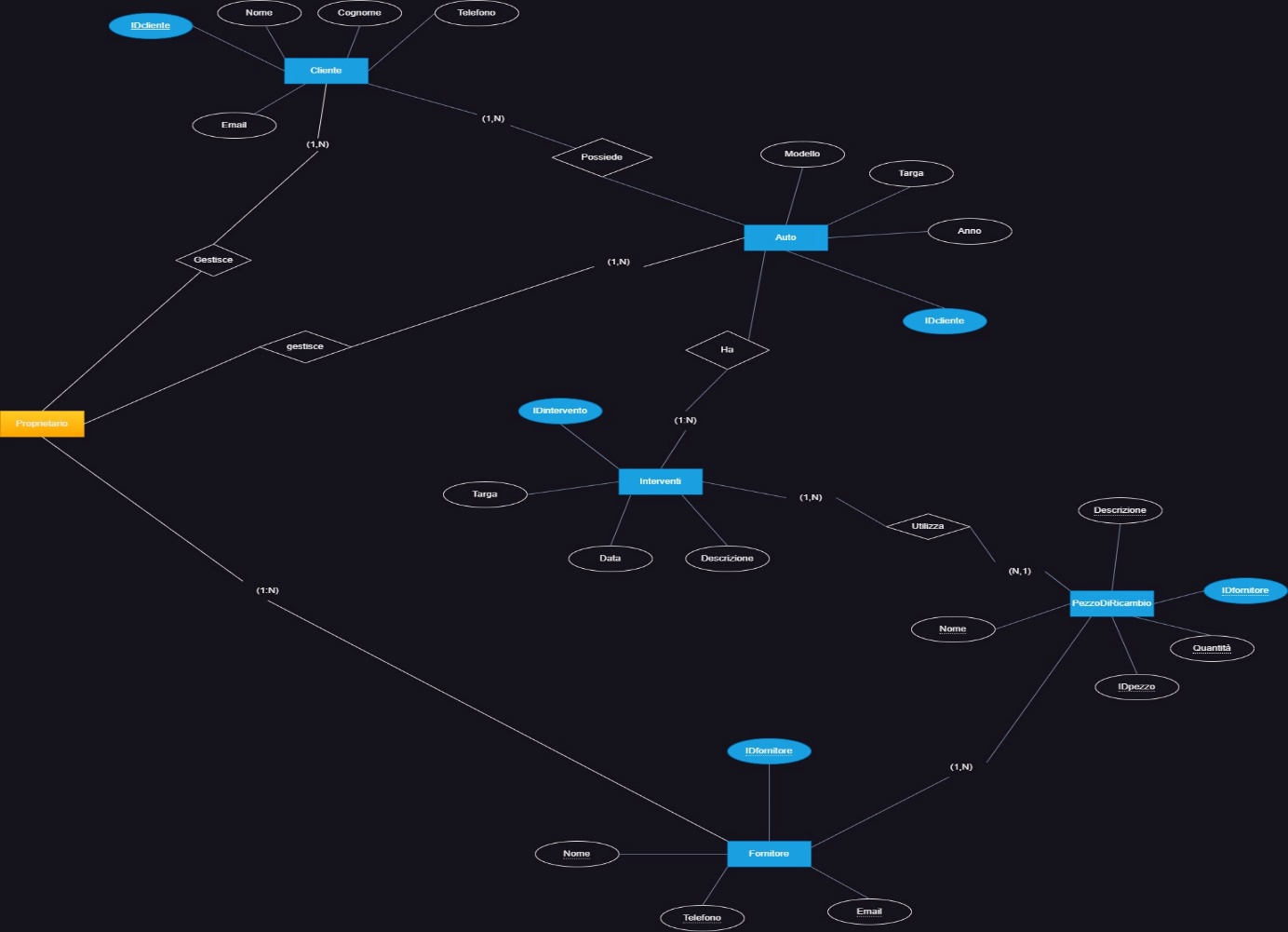
### **Descrizione dettagliata del modello E-R per il sistema di gestione dell'autofficina:**

#### **Entità e Attributi**

1. **Proprietario**
   * **ID Proprietario (PK)**
   * **Nome**
   * **Cognome**
   * **Username**
   * **Password**
   * **Nome Officina**
2. **Cliente**
   * **ID Cliente (PK)**
   * **Nome**
   * **Cognome**
   * **Email**
   * **Telefono**
3. **Auto**
   * **ID Auto (PK)**
   * **Marca**
   * **Modello**
   * **Anno**
   * **Targa**
   * **ID Cliente** (FK)
4. **Fornitore**
   * **ID Fornitore (PK)**
   * **Nome**
   * **Indirizzo**
   * **Telefono**
   * **Email**
5. **Pezzo di Ricambio**
   * **ID Pezzo (PK)**
   * **Nome**
   * **Descrizione**
   * **Prezzo**
   * **ID Fornitore** (**FK**)
6. **Intervento**
   * **ID Intervento (PK)**
   * **Data**
   * **Descrizione**
   * **Costo**
   * **ID Auto (FK)**
   * **ID Pezzo (FK**)

#### **Relazioni**

1. **Proprietario gestisce Auto**
   * Relazione uno a molti tra Proprietario e Auto
   * Un Proprietario può gestire più Auto
2. **Proprietario gestisce Cliente**
   * Relazione uno a molti tra Proprietario e Cliente
   * Un Proprietario può gestire più Clienti
3. **Cliente possiede Auto**
   * Relazione uno a molti tra Cliente e Auto
   * Un Cliente può possedere più Auto
4. **Proprietario gestisce Fornitore**
   * Relazione uno a molti tra Proprietario e Fornitore
   * Un Proprietario può gestire più Fornitori
5. **Fornitore fornisce Pezzi di Ricambio**
   * Relazione uno a molti tra Fornitore e Pezzo di Ricambio
   * Un Fornitore può fornire più Pezzi di Ricambio
6. **Auto ha Interventi**
   * Relazione uno a molti tra Auto e Intervento
   * Un'Auto può avere più Interventi
7. **Intervento utilizza Pezzo di Ricambio**
   * Relazione molti a uno tra Intervento e Pezzo di Ricambio
   * Un Intervento può utilizzare un solo Pezzo di Ricambio, ma un Pezzo di Ricambio può essere utilizzato in più Interventi



### **Descrizione delle Entità e delle Relazioni**

#### Proprietario

* Il **Proprietario** è l'entità principale che gestisce l'autofficina. Ha attributi personali come ID, Nome, Cognome, Username, Password e Nome dell'Officina.
* Il proprietario ha relazioni dirette con le entità **Cliente**, **Auto** e **Fornitore**.

#### Cliente

* Un **Cliente** è un individuo che possiede una o più auto. Ha attributi come ID, Nome, Cognome, Email e Telefono.
* Un cliente può possedere più **Auto**.

#### Auto

* Un'**Auto** è registrata nell'officina e ha attributi come ID, Marca, Modello, Anno e Targa.
* Un'auto è associata a un singolo cliente e può avere più **Interventi** registrati.

#### Fornitore

* Un **Fornitore** fornisce i pezzi di ricambio necessari per le riparazioni e ha attributi come ID, Nome, Indirizzo, Telefono ed Email.
* Un fornitore può fornire più **Pezzi di Ricambio**.

#### Pezzo di Ricambio

* Un **Pezzo di Ricambio** è una componente utilizzata negli interventi di riparazione. Ha attributi come ID, Nome, Descrizione, Prezzo e ID Fornitore.
* Un pezzo di ricambio può essere utilizzato in più **Interventi**.

#### Intervento

* Un **Intervento** è un'operazione di riparazione o manutenzione eseguita su un'auto. Ha attributi come ID, Data, Descrizione, Costo, ID Auto e ID Pezzo.
* Un intervento è associato a una singola auto e può utilizzare un singolo pezzo di ricambio, ma un pezzo di ricambio può essere utilizzato in più interventi.

### ***Conclusioni***

Il modello E-R descritto sopra fornisce una struttura dettagliata per il sistema di gestione di un'autofficina. Questo modello facilita la comprensione delle connessioni tra diverse entità e come esse interagiscono tra loro all'interno del sistema. La progettazione accurata delle relazioni e degli attributi garantisce una gestione efficiente dei dati e l'integrità referenziale nel database.

## **Traduzione dello schema concettuale in schema logico**

**Proprietario:**

* La tabella proprietario memorizza i dati del proprietario dell'autofficina.
* Le colonne includono informazioni personali come nome, cognome, username, password e il nome dell'officina.

**Cliente**

* La tabella cliente contiene i dati dei clienti.
* È collegata alla tabella proprietario tramite la colonna id\_proprietario, che è una chiave esterna che fa riferimento a proprietario(id\_proprietario).

**Auto**

* La tabella auto memorizza le informazioni sulle auto dei clienti.
* È collegata alla tabella cliente tramite la colonna id\_cliente, che è una chiave esterna che fa riferimento a cliente(id\_cliente).
* Inoltre, è collegata alla tabella proprietario tramite la colonna id\_proprietario.

**Fornitore**

* La tabella fornitore memorizza i dati dei fornitori.
* È collegata alla tabella proprietario tramite la colonna id\_proprietario.

**Pezzo di Ricambio**

* La tabella pezzo\_di\_ricambio contiene le informazioni sui pezzi di ricambio.
* È collegata alla tabella fornitore tramite la colonna id\_fornitore, che è una chiave esterna che fa riferimento a fornitore(id\_fornitore).

**Intervento**

* La tabella intervento memorizza i dati degli interventi di riparazione.
* È collegata alla tabella auto tramite la colonna id\_auto, che è una chiave esterna che fa riferimento a auto(id\_auto).
* Inoltre, è collegata alla tabella pezzo\_di\_ricambio tramite la colonna id\_pezzo, che è una chiave esterna che fa riferimento a pezzo\_di\_ricambio(id\_pezzo).

### **Conclusioni**

Questo schema logico rappresenta una traduzione dettagliata del modello concettuale in un formato relazionale, adatto per essere implementato in un database relazionale come MySQL. Le tabelle sono strutturate per garantire l'integrità referenziale e le relazioni tra le entità sono ben definite tramite chiavi esterne. Questo schema permette una gestione efficiente dei dati relativi a un'autofficina, garantendo coerenza e facilità di accesso alle informazioni.

# **Database implementation**

CREATE TABLE IF NOT EXISTS users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(255) NOT NULL,

email VARCHAR(100) NOT NULL,

officina\_name VARCHAR(100) NOT NULL,

reset\_code VARCHAR(255) NULL,

UNIQUE (username), UNIQUE (email) );

CREATE TABLE cliente (

id\_cliente INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(50) NOT NULL,

cognome VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL,

telefono VARCHAR(20) NOT NULL,

id\_proprietario INT,

FOREIGN KEY (id\_proprietario) REFERENCES proprietario(id\_proprietario)

);

CREATE TABLE auto (

id\_auto INT AUTO\_INCREMENT PRIMARY KEY,

marca VARCHAR(50) NOT NULL,

modello VARCHAR(50) NOT NULL,

anno YEAR NOT NULL,

targa VARCHAR(20) NOT NULL UNIQUE,

id\_cliente INT,

FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente),

id\_proprietario INT,

FOREIGN KEY (id\_proprietario) REFERENCES proprietario(id\_proprietario)

);

CREATE TABLE fornitore (

id\_fornitore INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

indirizzo VARCHAR(255) NOT NULL,

telefono VARCHAR(20) NOT NULL,

email VARCHAR(100) NOT NULL,

id\_proprietario INT,

FOREIGN KEY (id\_proprietario) REFERENCES proprietario(id\_proprietario)

);

CREATE TABLE pezzo\_di\_ricambio (

id\_pezzo INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

descrizione TEXT NOT NULL,

prezzo DECIMAL(10, 2) NOT NULL,

id\_fornitore INT,

FOREIGN KEY (id\_fornitore) REFERENCES fornitore(id\_fornitore)

);

CREATE TABLE intervento (

id\_intervento INT AUTO\_INCREMENT PRIMARY KEY,

data DATE NOT NULL,

descrizione TEXT NOT NULL,

costo DECIMAL(10, 2) NOT NULL,

id\_auto INT,

FOREIGN KEY (id\_auto) REFERENCES auto(id\_auto),

id\_pezzo INT,

FOREIGN KEY (id\_pezzo) REFERENCES pezzo\_di\_ricambio(id\_pezzo)

);

**Diagramma dei Casi d’Uso**

I diagrammi dei casi d’uso descrivono le interazioni tra attori (utenti) e il sistema. Ora vedremo i principali casi d’uso per il sistema di gestione dell’autofficina.

1. Registrazione del Proprietario

* Descrizione: il proprietario dell’autofficina si registra nel sistema
* Attore: proprietario
* Passi:

1. Il proprietario inserisce il nome utente, password e il nome dell’officina
2. Il sistema registra le informazioni sul database
3. Login

* Descrizione: il proprietario accede al sistema
* Attore: proprietario
* Passi:

1. Il proprietario inserisce nome utente e password
2. Il sistema verifica le credenziali e concede l’accesso
3. Registrazione Clienti

* Descrizione: il proprietario registra nuovi clienti nel sistema
* Attore: proprietario
* Passi:

1. Il proprietario inserisce le informazioni del cliente
2. Il sistema registra le informazioni del cliente nel database
3. Registrazione Auto

* Descrizione: il proprietario registra nuove auto nel sistema
* Attore: proprietario
* Passi:

1. Il proprietario inserisce le informazioni dell’auto e il cliente associato
2. Il sistema registra le informazioni dell’auto nel database
3. Registrazione Fornitori

* Descrizione: il proprietario registra nuovi fornitori nel sistema
* Attore: proprietario
* Passi:

1. Il proprietario inserisce le informazioni del fornitore
2. Il sistema registra le informazioni del fornitore nel database
3. Registrazione Pezzi di Ricambio

* Descrizione: il proprietario registra nuovi pezzi di ricambio nel sistema
* Attore: proprietario
* Passi:

1. Il proprietario inserisce le informazioni del pezzo di ricambio e il fornitore associato
2. Il sistema registra le informazioni del pezzo di ricambio nel database
3. Registrazione Interventi

* Descrizione: il proprietario registra nuovi interventi nel sistema
* Attore: proprietario
* Passi

1. Il proprietario inserisce le informazioni dell’intervento, inclusi auto e pezzi di ricambio
2. Il sistema registra le informazioni dell’intervento nel database

## Immagine che contiene diagramma, testo, linea, bianco e nero Descrizione generata automaticamente

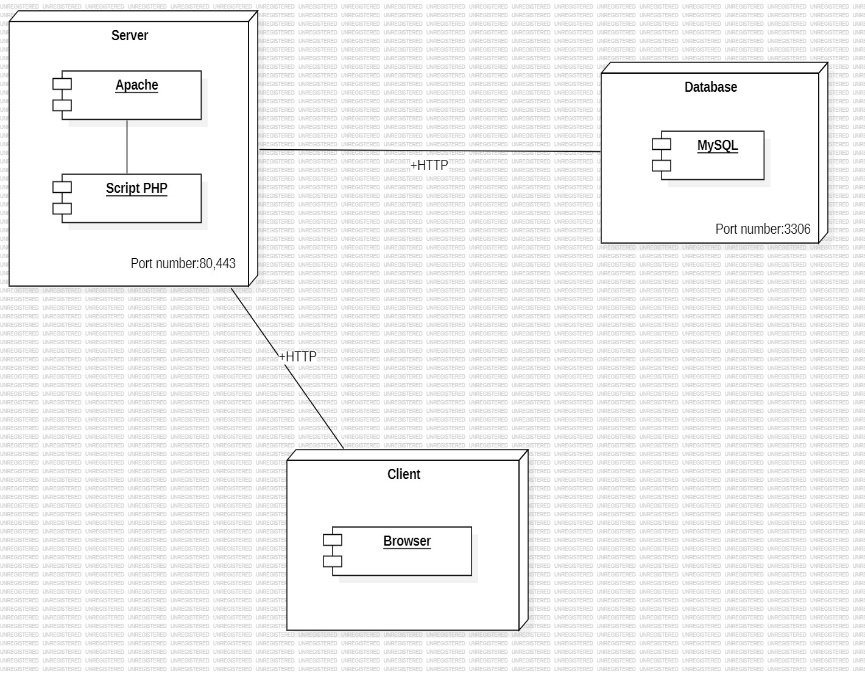
## Software Interfaces

Per lo sviluppo di questa web application è stato deciso di utilizzare i seguenti linguaggi:

* PHP come tecnologia server-side per la gestione dei dati inviati/letti al/dal database.
* HTML per la presentazione dei dati.
* CSS per la formattazione dello stile dell'applicazione.
* Javascript per validazione dei dati e visibilità dinamica dei campi di inserimento

## Communications Interfaces

L’infrastruttura sulla quale poggia il sistema è costituita dal web server Apache che gestisce le richieste del client web(browser) e recupera le risorse richieste dal database. MySql è l’RDBMS che si occupa della gestione del database. La comunicazione avviene tramite scambio di messaggi HTTP.



# **System Features**

Il **Sistema di Accesso** per un'autofficina è una funzionalità cruciale che permette al proprietario di accedere alle funzionalità gestionali del sistema. Di seguito viene fornita una descrizione dettagliata del **System Feature Accesso**, che include registrazione, login e gestione delle sessioni.

 **Registrazione del Proprietario**

* Il sistema deve consentire la registrazione di un nuovo proprietario dell'autofficina.
* I dettagli necessari per la registrazione includono: username, password e nome dell'officina.
* Il sistema deve garantire che l'username sia univoco.

 **Login**

* Il sistema deve permettere al proprietario registrato di effettuare il login utilizzando l'username e la password.
* In caso di credenziali errate, il sistema deve notificare l'errore all'utente.

 **Gestione delle Sessioni**

* Il sistema deve mantenere una sessione attiva per l'utente autenticato.
* Deve esserci la possibilità di effettuare il logout, terminando così la sessione attiva.
* Il sistema deve reindirizzare gli utenti non autenticati alla pagina di login.

 **Reindirizzamento**

* Dopo un login avvenuto con successo, il sistema deve reindirizzare il proprietario alla pagina di inserimento dati.
* Dopo la registrazione, il sistema deve reindirizzare alla pagina di login.