



SISTEMA DI CONTROLLO CENTRALE DI UN AUTOVEICOLO

Realizzato da
Halyna Zamorska
Carminé Palo



OBIETTIVI



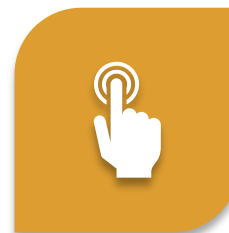
**CONSENTIRE LA
SUPERVISIONE DELLA PARTE
MECCANICA DEL VEICOLO
ATTRAVERSO L'USO DI
APPOSITI SENSORI**



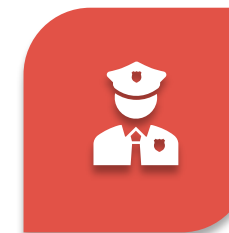
**CONSENTIRE LA LETTURA
SEQUENZIALE DI DATI
ATTRAVERSO L'USO DI UN
DATABASE**



**RENDERE TALE LETTURA PIU'
ACCESSIBILE ATTRAVERSO LA
CREAZIONE DI UN'
INTERFACCIA CLIENT**

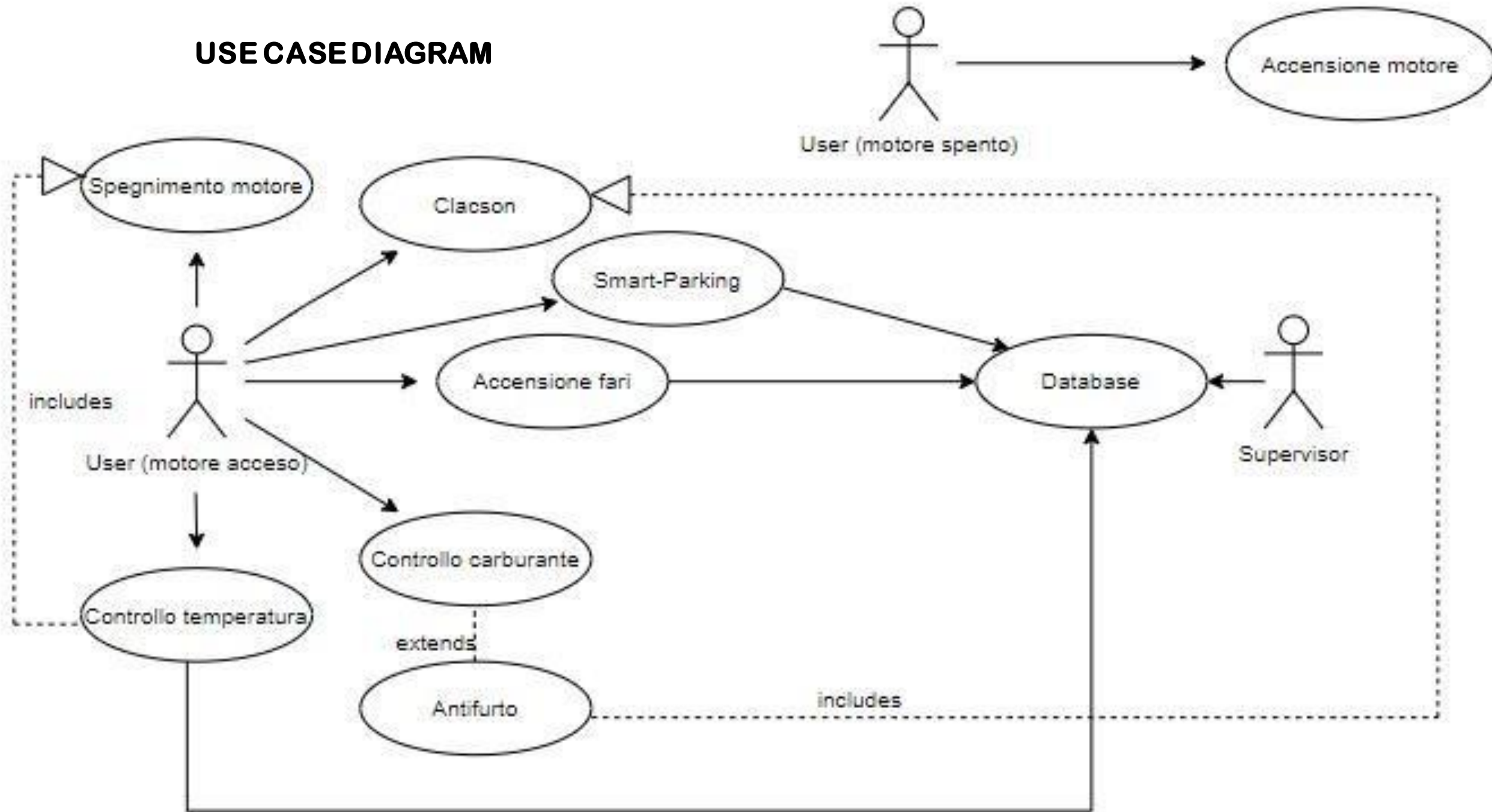


**CREARE UN CLIENT WEB CHE
PERMETTE ALL'UTENTE
FINALE DI DIALOGARE CON LA
PARTE MECCANICA DEL
MEZZO**

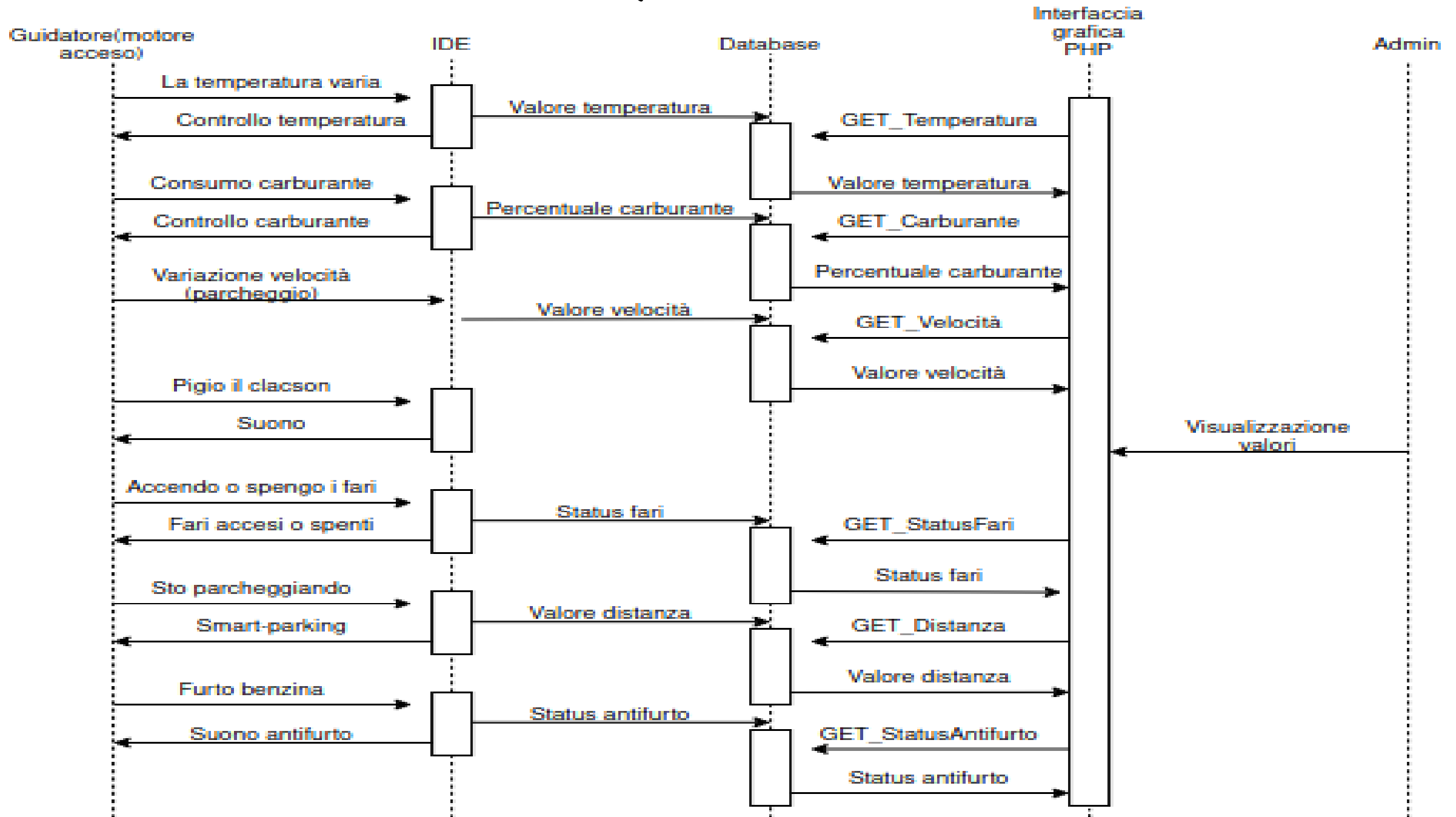


**CREARE UN SISTEMA DI
PROTEZIONE DA INCENDIO E
FURTO DI BENZINA**

USE CASE DIAGRAM



SEQUENCE DIAGRAM



PROGETTAZIONE ARCHITETTURALE

Il progetto in questione prevede:

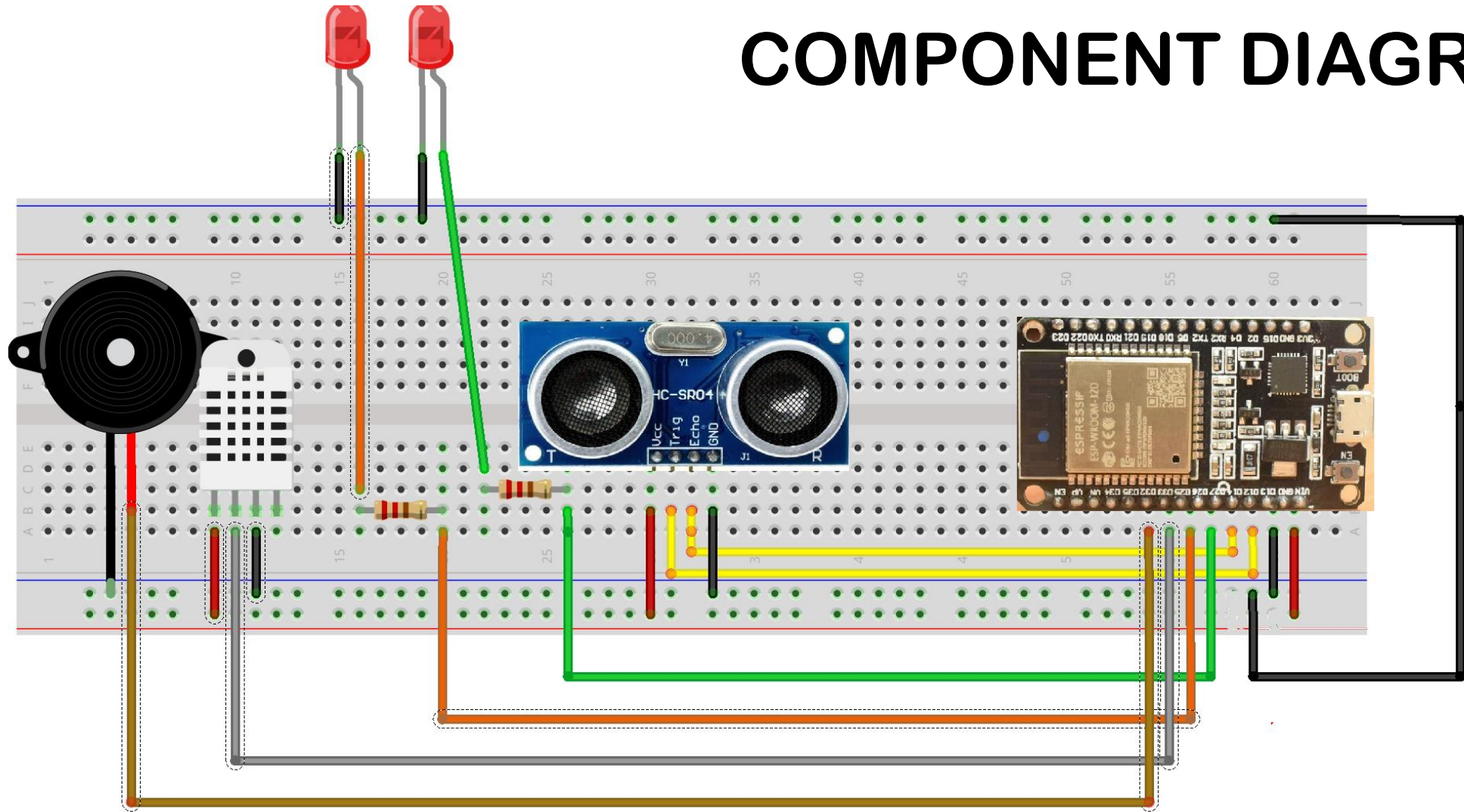
- Una parte hardware (esp32-WROOM 32 ,varia sensoristica)
- Tre pagine formato php (Per la simulazione lato Utente e per la ricezione e l'invio di dati al database)
- Un database (Deposito dati , costruito usando phpmyadmin)
- Arduino IDE (Per la programmazione dell'ESP-32)

DESCRIZIONE HARDWARE

Nella progettazione del sistema fisico, alimentato tramite USB da computer, sono stati usati:

- . **Esp-32** (è una serie di sistemi a basso costo e bassa potenza su microcontrollori con chip con Wi-Fi integrato e Bluetooth dual-mode)
- . **Breadboard**
- . **Sensoristica varia** (Sensori di temperatura ed ultrasuoni)
- . **Componenti elettronici** (resistori ,led o componenti con output acustico come quello utilizzato per la simulazione del clacson)

COMPONENT DIAGRAM



Per praticità la scheda ESP32 è alimentata tramite il cavo usb di un personal computer, ma puo' essere alimentata tramite altre fonti portatili.

ARDUINO IDE

Grazie a questo strumento specializzato per i sistemi arduino , è stato possibile programmare l'ESP-32. Il programma è composto da 4 macrosezioni:

- Sezione iniziale (Inclusione librerie , configurazione Wi-Fi , dichiarazione dei pin ed altre variabili successivamente utili)
- Setup (Configurazione dei pin , connessione Wi-Fi)
- Loop e programmazione funzioni

Sono presenti i prototipi delle seguenti funzioni poi implementate:

- **ControlloTemperatura** che rileva e trasmette i dati della temperatura al database
- **ProtezioneFurto** che aziona il clacson se il livello del carburante si abbassa troppo velocemente

- **CalcoloDistanza** che serve per rilevare gli ostacoli dal sensore ad ultrasuoni e a seconda della distanza rilevata aumenta o diminuisce la frequenza dei suoni del sensore di parcheggio
- **CalcoloVelocitaDiSpostamento** viene calcolata la velocità con cui il veicolo si avvicina o si allontana dagli ostacoli considerando il sensore di parcheggio
- **ControlloBenzina** che aggiorna la quantità di benzina attuale in real-time e spegne il veicolo nel caso dell'esaurimento del carburante
- **ConnessioneInsertDb** che inserisce i dati nel database
- **ProtezioneSurriscaldamento** che avvisa l'utente nel caso del superamento della temperatura di sicurezza per il sistema
- **AggiornamentoSensori** che serve per tenere traccia in ogni istante dello status dei sensori, nel caso in cui una funzione ne cambia lo status, vengono aggiornate fisicamente da qui

CODICE PIU' SIGNIFICATIVO

```
void distanzaSicurezza(){
float spostamento;
float conversione;

if(distanza <60){
    if(distanza1 == 0){
        distanza1 = distanza;
        Serial.print("distanza 1 e' ");
        Serial.println(distanza1);
    }
    else{
        distanza2 = distanza;
        Serial.print("distanza 2 e' ");
        Serial.println(distanza2);
    }

    if(distanza1 != 0 & distanza2 != 0){
        spostamento = distanza2 - distanza1;
        // spostamento = spostamento / 100; //passo dai centimetri ai metri

        Serial.print("Il valore della var spostamento e' ");
        Serial.println(spostamento);
        conversione = ritardoLoop/1000; //passo dai millisecondi ai secondi

        Serial.print("Il valore della var conversione e' ");
        Serial.println(conversione);
        velocita = spostamento/conversione;

        Serial.print ("La velocità rispetto alla distanza è= ");
        Serial.print (velocita);
        Serial.println ("centimetri al secondo");
        distanza1 = 0;
        distanza2 = 0;
    }
}
```

```
void connessioneInsertDb(){

    if (client.connect(serverName, 80)) {

        Serial.println("connected");

        // Make a HTTP request:

        Serial.print("GET /esempiPhp/insert_mysql2.php?Temperatura=");

        client.print("GET /esempiPhp/insert_mysql2.php?Temperatura="); //YOUR URL

        Serial.println(temperatura);

        client.print(temperatura);

        client.print("&statusVeicolo=");

        Serial.println("&statusVeicolo=");

        client.print(statoGENERALE);

        Serial.println(statoGENERALE);

        client.print("&statusFari=");

        Serial.println("&statusFari=");

        client.print(LED1status);

        Serial.println(LED1status);

        client.print("&benzina=");
    }
}
```

```

float ritardoMin = 3;

long ritardo = 0;

float m = (ritardoMax-ritardoMin)/(distanzaMax-distanzaMin);
// imposta l'uscita del trigger LOW

digitalWrite(pinTrigger, LOW);

// imposta l'uscita del trigger HIGH per 10 microsecondi

digitalWrite(pinTrigger, HIGH);

delayMicroseconds(10);

digitalWrite(pinTrigger, LOW);

// calcolo del tempo attraverso il pin di echo

long durata = pulseIn(pinEcho, HIGH);

distanza = durata/58.31;

Serial.print("distanza: ");

Serial.println(distanza);

if(distanza<60){
    ritardo = m*distanza;
digitalWrite(pinBuzzer,HIGH);
delay(ritardo);
digitalWrite(pinBuzzer,LOW);
delay(ritardo);

```

```

}

void handle_ledon() {

if(fuel != 0){

    LED1status = HIGH;
    LED2status = HIGH;

    Serial.println(" LED1status e LED2status sono high");
    server.send(200, "text/html", SendHTML(true,true,CLACSONstatus,BENZINAsstatus,statoGENERALE,furtoBenzina));

}else{
    Serial.println("E' finita la benzina");
}

}

void handle_ledoff() {

    LED1status = LOW;
    LED2status = LOW;
    Serial.println("LED1status e LED2status sono low");
    server.send(200, "text/html", SendHTML(false,false,CLACSONstatus,BENZINAsstatus,statoGENERALE,furtoBenzina));
}

void handle_clacsonon() {
    CLACSONstatus = HIGH;

    Serial.println(" CLACSONstatus e' HIGH");
    server.send(200, "text/html", SendHTML(LED1status,LED2status,true,BENZINAsstatus,statoGENERALE,furtoBenzina));
}

void handle_clacsonoff() {

```

PAGINE PHP

PHP pagina interazione con l'utente:

Chiamata interamente dall'Arduino Ide , **sfrutta le potenzialità di HTML+CSS con lo scopo di creare un insieme di bottoni pensati per interagire con la parte meccanica.** Ognuna delle funzionalità descritte ha una pagina HTML corrispondente che una volta aperta chiama le funzioni implementate sull'ide, in particolare sono:

1. Accensione/spegnimento fari
2. Azionamento del clacson
3. Azionamento del veicolo
4. Rifornimento del carburante

Nelle righe di codice a seguire , la stessa pagina **si occupa dell'inserimento delle informazioni nel database associato chiamando una seconda pagina php che si occupa esclusivamente degli inserimenti.**

-Pagina PHP per la visualizzazione dei dati:

Questa pagina ,attraverso l'uso del comando "GET" , ha lo scopo di stampare a video tutti i dati depositati nel database.

Nelle prime righe di codice è possibile notare la funzione che **ha il compito di collegarsi al database per poi prelevarne le informazioni.** Con il comando "ECHO" , infine , stamperà le variabili indicate a video. Come la precedente pagina descritta , anche qui è stato usato il CSS (linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML, ad esempio i siti web e relative pagine web) per offrire una migliore visualizzazione dei dati.

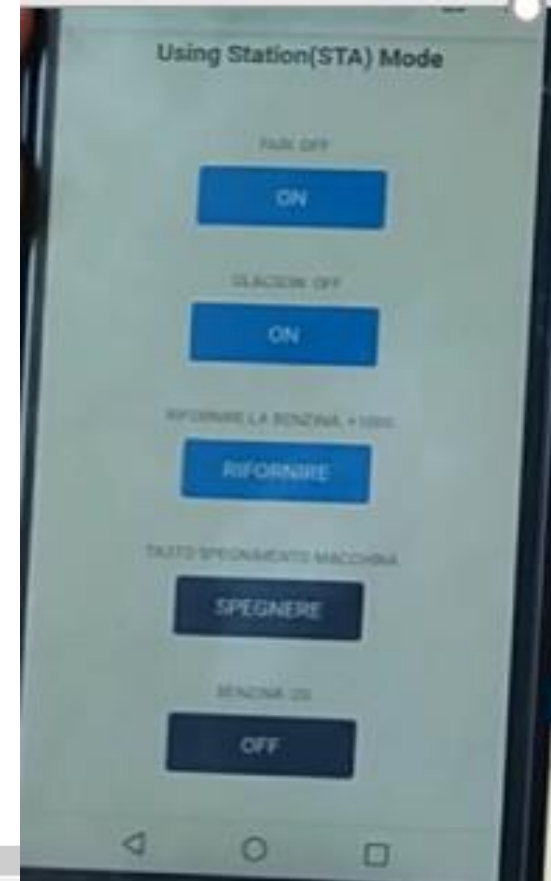

```
String SendHTML(uint8_t led1stat,uint8_t led2stat,uint8_t clacsonstat,uint8_t benzinastat,uint8_t attivazionestat,uint8_t furtoBenzina){
    String ptr = "<!DOCTYPE html> <html>\n";
    ptr +="<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";
    ptr +="<title>Control Server</title>\n";
    ptr +="<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";
    ptr +="body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;} h3 {color: #444444;margin-bottom: 50px;}\n";
    ptr +=".button {display: block;width: 80px;background-color: #3498db;border: none;color: white;padding: 13px 30px;text-decoration: none;font-size: 16px;margin: 0px auto 35px;cursor: pointer;border-radius: 4px\n";
    ptr +=".button-on {background-color: #3498db;}\n";
    ptr +=".button-on:active {background-color: #2980b9;}\n";
    ptr +=".button-off {background-color: #34495e;}\n";
    ptr +=".button-off:active {background-color: #2c3e50;}\n";
    ptr +="p {font-size: 12px;color: #888;margin-bottom: 10px;}\n";
    ptr +="</style>\n";
    ptr +="</head>\n";
    ptr +="<body>\n";
    ptr +="<h1>ESP32 Web Server</h1>\n";
    ptr +="<h3>Using Station(STA) Mode</h3>\n";

    if(led1stat&led2stat)
    {ptr +="<p>FARI: ON</p><a class=\"button button-off\" href=\"/ledoff\">OFF</a>\n";}
    else
    {ptr +="<p>FARI: OFF</p><a class=\"button button-on\" href=\"/ledon\">ON</a>\n";}

    if(clacsonstat)
    {ptr +="<p>CLACSON: ON</p><a class=\"button button-off\" href=\"/clacsonoff\">OFF</a>\n";}
    else
    {ptr +="<p>CLACSON: OFF</p><a class=\"button button-on\" href=\"/clacsonon\">ON</a>\n";}

    if(benzinastat)
    {ptr +="<p>BENZINA RIFORNITA: +1000 </p><a class=\"button button-off\" href=\"/rifornimentoOff\">RIFORNITA</a>\n";}
    else
    {ptr +="<p>RIFORNIRE LA BENZINA: +1000 </p><a class=\"button button-on\" href=\"/rifornimentoOn\">RIFORNIRE</a>\n";}

    if(attivazionestat)
    {ptr +="<p>ATTIVAZIONE: ON</p><a class=\"button button-off\" href=\"/attivazioneOff\">OFF</a>\n";}
    else
    {ptr +="<p>ATTIVAZIONE: OFF</p><a class=\"button button-on\" href=\"/attivazioneOn\">ON</a>\n";}
```



```

html>
<head><title>Lettura Database</title></head>

<body>
  <?php

    echo "<p style = 'font-size:30px;margin:16px;'>". "Benvenuti nel sistema di monitoraggio digitale  </p>";

    echo "<p style = 'font-size:24px;margin:16px;'>". "Istruzioni d'utilizzo  </p>";

    echo "<p style = 'font-size:20px;margin:16px;'>". "Il seguente sistema preleva
    letture presenti nel database della centralina di un autoveicolo. <br>Queste informazioni sono
    tutti quei dati che stanno trasmettendo in real-time i sensori installati sull'autoveicolo.  </p><br>";

echo "<span ><table width='1000' border = '2px solid black'>";

    echo "<tr>";

    echo "<td width='110' height='58' style = 'font-size:24px;text-align:center;background-color:#7fffd4; border:1px solid grey;'>№ record</td>";
    echo "<td width='110' height='58' style = 'font-size:24px;text-align:center;background-color:#F5F5DC; border:1px solid grey;'>Temperatura del motore</td>";
    echo "<td width='110' height='58' style = 'font-size:24px;text-align:center;background-color:#Ffe4c4; border:1px solid grey;'>Veicolo spento o acceso,1 significa acceso, 0 spento</td>";
    echo "<td width='110' height='58' style = 'font-size:24px;text-align:center;background-color:#F5F5DC; border:1px solid grey;'>Fari spenti o accesi</td>";
    echo "<td width='110' height='58' style = 'font-size:24px;text-align:center;background-color:#Ffe4c4; border:1px solid grey;'>Quantita di benzina </td>";
    echo "<td width='110' height='58' style = 'font-size:24px;text-align:center;background-color:#F5F5DC; border:1px solid grey;'>Distanza tra il veicolo e altri ostacoli</td>";
    echo "<td width='110' height='58' style = 'font-size:24px;text-align:center;background-color:#Ffe4c4; border:1px solid grey;'>Velocita di spostamento del veicolo</td>";
    echo "<td width='110' height='58' style = 'font-size:24px;text-align:center;background-color:#F5F5DC; border:1px solid grey;'>Stato di antifurto</td>";
    //echo "<td width='35' height='58'><a href='cart.php?action=delete&ship_id=$cartId'><img src='images/trash.png' width='31' height='42' /></a></td>";
    //echo "<td height='58'>$cartId</td>";
    echo "</tr>";
    //}
    echo "</table></div>";

    $myconn = mysqli_connect("localhost","root","","nuovo");
    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT * FROM arduino"; //Seleziona tutto dalla tabella del db

```

DATABASE

-Il database è stato progettato usando la funzionalità "**Phpmyadmin**" inclusa nel programma "**xampp**". **Phpmyadmin** offre un' interfaccia comoda e di facile utilizzo atta alla creazione , visualizzazione o modifica del database rispetto alla gestione classica da linea di comando.

Il database è formato da una sola tabella perché deposita i dati per una sola centralina. Nel caso di più sistemi centralizzati , il database **può essere ampliato** usando più tabelle utilizzando in modo opportuno parametri quali la "**primary key**" o la "**foreign key**" per l'identificazione dei vari sistemi.

Un' apposita pagina php senza un' interfaccia grafica e' responsabile dell'inserimento dei dati nel database e viene chiamata direttamente dalla parte meccanica del sistema.