

Design and implementation of a vehicle geolocation system using Arduino Mega2560, Gsm / Gps Sim808 and Esp 8266 microcontrollers

Pinatibi Hyann¹, Soro Mamadou¹, Ouattara Gninnafanlan Justin¹, and Coulibaly Talnan Jean Honoré²

¹Department of Fundamental and Applied Sciences, mathematics and computer science Laboratory, Nangui Abrogoua University, Abidjan, Côte d'Ivoire

²Department of Environmental Sciences and Management, Geosciences and environment Laboratory, Nangui Abrogoua University, Abidjan, Côte d'Ivoire

Copyright © 2026 ISSR Journals. This is an open access article distributed under the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT: This study involved setting up a real time vehicle tracking system using the Arduino Mega 2560 board. This solution will be useful for anyone who wants to track their vehicle's movements in real time. The aim of this project is to optimize vehicle geolocation by combining several geolocation methods, including GSM and GPS based geolocation. This system consists of two parts: a hardware component built using the Arduino board and its GPS / GSM SIM 808 and ESP 8266 modules and a software component built using the Laravel Framework to process data from the hardware system. The SIM 808 GPS / GSM module with a GPS sensitivity of -165 dBm has a good performance in challenging environments like urban or indoors places. This module enabled vehicle tracking with a speed accuracy (DGPS) of 0.05 m / s, an acceleration accuracy (DGPS) of 0.05 m / s² and a time accuracy of 10 ns, in this study test speeds ranging from 35 km / h to 65 km / h. Data transmission and reception by the ESP8266 module occurs at a frequency of 2.4 GHz and a network transmission signal level of +19.5 dBm.

KEYWORDS: Geolocation, Microcontroller, GSM, Arduino, Embedded system.

1 INTRODUCTION

The global number of vehicles is expected to increase as ownership becomes more affordable thanks to economic growth in developing countries [1]. On the other hand, car thefts have also increased at an alarming rate around the world. As in many other countries, numerous cases of vehicle theft are reported in Côte d'Ivoire, and this figure is increasing every year [2]. It is therefore essential to put security systems in place to combat these various sources of insecurity associated with vehicle ownership. In order to limit theft, vehicles are equipped with basic locks by manufacturers. However, these locks are not sufficient to secure the vehicle. Given the vulnerability of locks, vehicle tracking systems that can determine the location of a vehicle at any time are being designed by technology companies. The main components of these systems are GPS (Global Positioning System) tags [3], [4], [5]. Many researchers have addressed GPS-related work [6], [7], [8], [9], [10], [11], [12], [13]. However, the deployment of GPS / GSM systems is still very limited on the market, as commercially available vehicle tracking anti-theft systems are relatively expensive and some studies estimate that the GPS method consumes a lot of energy [14], [15]. As an alternative, microcontrollers and associated modules are having an increasingly significant impact in today's environment. ARDUINO microcontrollers and Raspberry Pi single-board computers are the brains behind Internet of Things (IoT) systems, which enable real-time data collection and processing [16]. Indeed, the brains of IoT systems, such as Arduino microcontrollers or others, single-board computers (Raspberry Pi) and the various modules connected to these processing units (Arduino, Raspberry Pi) are less expensive and consume very little energy [17]. IoT systems are controlled by single-board computers with up to eight gigabits of RAM, similar to those found in ordinary computers and have Wi-Fi and low-energy Bluetooth-based internet access, enabling easy interconnection and data transmission [18]. IoT based vehicle geolocation systems consist of a microcontroller or single-board computer, a GPS/GSM module for collecting the vehicle's geographical coordinates (GPS) and transmitting messages to the driver (GSM) and an ESP module that connects to a web application to

record information from all cars connected to the location system and to track the vehicle's various positions on a Google map [19]. This means that an administrator should be able to track the car by viewing the coordinates on a map and checking the driver's location in real time using a computer. IoT-based geolocation systems offer reliability, adaptability, scalability and accessibility provided by cloud servers [20]. This study focuses on setting up an IoT-based geolocation system. The Arduino MEGA2560 microcontroller will be used for real-time data processing, while the SIM 808 GSM / GPS module will be used for geolocation and communication with the driver in order to retrieve the various geographical coordinates of the vehicles and display them on a Google map. The link between the hardware aspects of the system, namely the Arduino MEGA2560 microcontroller, the SIM808 GSM/GPS module and the web-based location applications (Google Maps) and web application used to record various information (make, color, license plate number, etc.) about the vehicles managed by our system, will be provided by the ESP 8266 module.

2 MATERIALS AND METHODS

2.1 MATERIALS

The implementation of the project requires the use of the following equipment: an Arduino MEGA2560 microcontroller, an ESP8266 module, a website, and a SIM808 GSM/GPS module. Their various characteristics will be summarized in the following sections.

2.1.1 ARDUINO MEGA 2560 MICROCONTROLLER

The Arduino Mega 2560 module (Fig. 1.) is based on the ATmega 2560 microcontroller. The Arduino Mega 2560 has 54 digital input / output pins, 14 of which can be used as PWM (Pulse Width Modulation) outputs numbered 2 to 13 and 44 to 46 (optional), 16 analog pins numbered A0 to A15, and four serial communication pins between the hardware: pins 0 (RX) and 1 (TX), 19 (RX) and 18 (TX), 17 (RX) and 16 (TX), and 15 (RX) and 14 (TX). The Arduino Mega 2560 module also has a 16 MHz quartz oscillator as its processing frequency, a USB connection for communicating with a computer, a power supply socket, and an ICSP connector for programming the Arduino board, as well as for connecting two or more Arduino boards. The ATmega 2560 microcontroller has 256 KB of FLASH memory for storing the program, 8 KB of SRAM (volatile) memory and 4 KB of EEPROM (non-volatile, electrically erasable memory, readable using the EEPROM library). The Arduino Mega 2560 microcontroller uses programs written in C language to perform real-time processing of data collected using sensors in an IoT system.

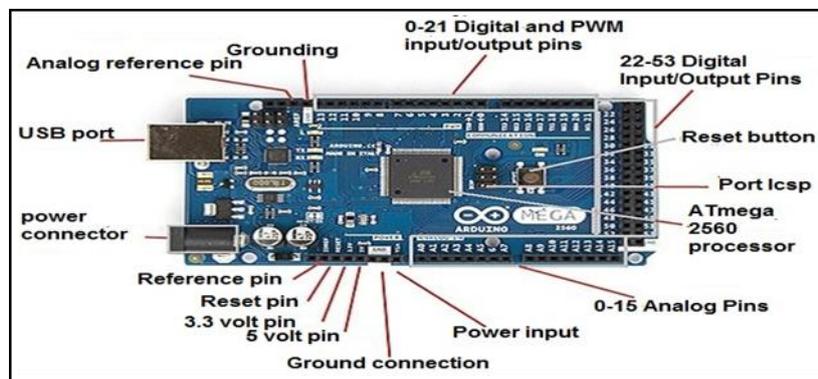


Fig. 1. Arduino Mega 2560 microcontroller

2.1.2 ESP 8266 WIFI MICROCONTROLLER

The ESP8266 (Fig. 2) is a popular and widely used Wi-Fi microcontroller developed by Espressif Systems. It integrates a 32-bit Tensilica microcontroller (MCU) and a Wi-Fi module, making it ideal for Internet of Things (IoT) applications. The ESP8266 module provides Wi-Fi connectivity, allowing devices to connect wirelessly to local networks or the Internet. It supports various Wi-Fi protocols, including 802.11b/g/n. Specifically, 802.11b is characterized by a throughput of 11 Mbps on the 2.4 GHz band, 802.11g is characterized by a throughput of 54 Mbps on 2.4 GHz and 802.11n is characterized by a throughput of up to 600 Mbps on the 2.4 GHz and 5 GHz bands. The ESP8266 incorporates a powerful 32-bit microcontroller based on the Xtensa LX106 architecture. It operates at clock frequencies ranging from 80 to 160 MHz and offers a range of GPIO pins for interfacing with external components. It also has built-in flash memory for storing micro software and programming code. The ESP8266 can be

programmed with the Arduino IDE or other popular programming platforms such as Micro Python and NodeMCU. It has an extensive ecosystem of libraries that allow developers to easily create applications and connect to various sensors, actuators, and other devices. The ESP8266 is designed to operate efficiently, making it suitable for battery-powered or low-power applications. It offers sleep modes and power management features to optimize power consumption. The ESP8266 has found applications in various IoT projects, such as home automation, smart appliances, sensor networks, weather stations, industrial monitoring, and more. Its affordable price, ease of use, and robustness make it a popular choice among hobbyists, makers, and professional developers.

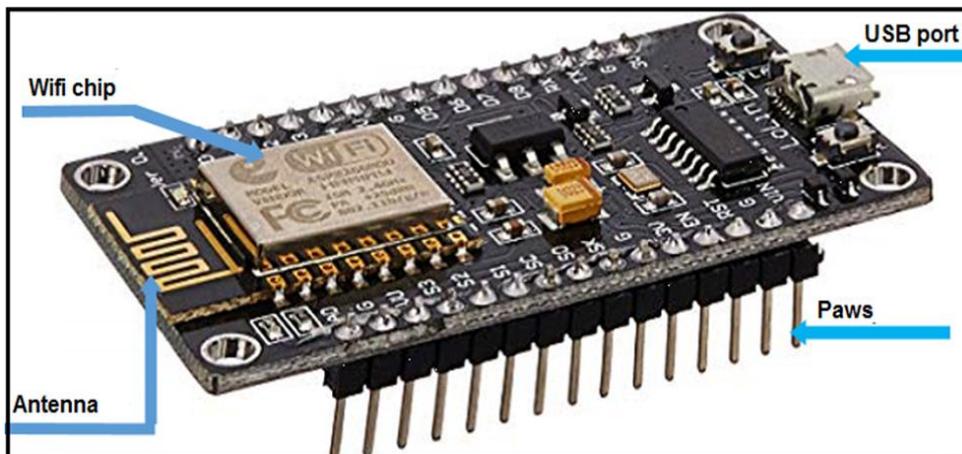


Fig. 2. Esp 8266 Wifi microcontroller

2.1.3 GSM / GPS SIM 808 MICROCONTROLLER

Arduino GSM/GPS module (Fig. 3) is a microcontroller, programmable cards that can connect to the mobile phone network. It allows you to make and receive phone calls, text messages, and multimedia messages. To program such a module and exchange information with it, you combine it with an Arduino card. These boards usually come with a GPS antenna when purchased. The GSM module is capable of sending and receiving mobile phone information. The GPS antenna is used for geolocation. In the case of vehicle tracking, for example, the system acquires the vehicle's positions via the GPS receiver and then sends the data to the monitoring center via SMS (Short Message Services) or GPRS (General Package Radio Service). They have input and output pins through which they receive commands and share the information collected. The entire system is controlled by a SIM card that is inserted into the module, hence their name beginning with SIM. There are the older SIM8080s and the newer SIM900s, which can be found in Côte d'Ivoire in the online store called centronick ci. As an advantage, this module has a small size of 54 * 42mm and is available worldwide. It has perfectly met the needs of the wireless data transmission and positioning function. We can see the Gsm / Gps Sim 808 general specification in (Table 1).



Fig. 3. Gps / Gsm Sim 808 microcontroller

Table 1. Gps / Gsm Sim 808 general specification

General Specification	
Chipset	SIM808 (GSM / GPRS / GPS)
Network Band	Quad-Band 850 / 900 / 1800 / 1900 Mhz
GPS Receiver	MT3337
GPS Sensitivity	-165 dBm
Operating Voltage	3.1 V – 4.8 V DC
Current Consumption	1 mA (Sleep Mode)
Interface	UART (Serial Communication)
Functions	SMS, Call, GPRS, GPS, Tracking
Compatibility	Arduino, Raspberry Pi, ESP32, STM32
Antenna Support	GSM and GPS External
Dimensions	Approx. 45 mm x 45 mm x 15 mm
Operating Temperature	-40 C to + 85 C
Shipment Weight	0.045 Kg
Shipment Dimensions	6x5x3 cm

2.1.4 ARDUINO IDE

The Arduino integrated development environment (Fig. 4.) is a cross-platform application written in C and C++. It is used to write and upload programs to Arduino-compatible boards. A program is called a sketch. The interface consists mainly of a program input area (1), a button to move or upload the program to the Arduino board (2), and another button to view the program’s progress on a computer (3).

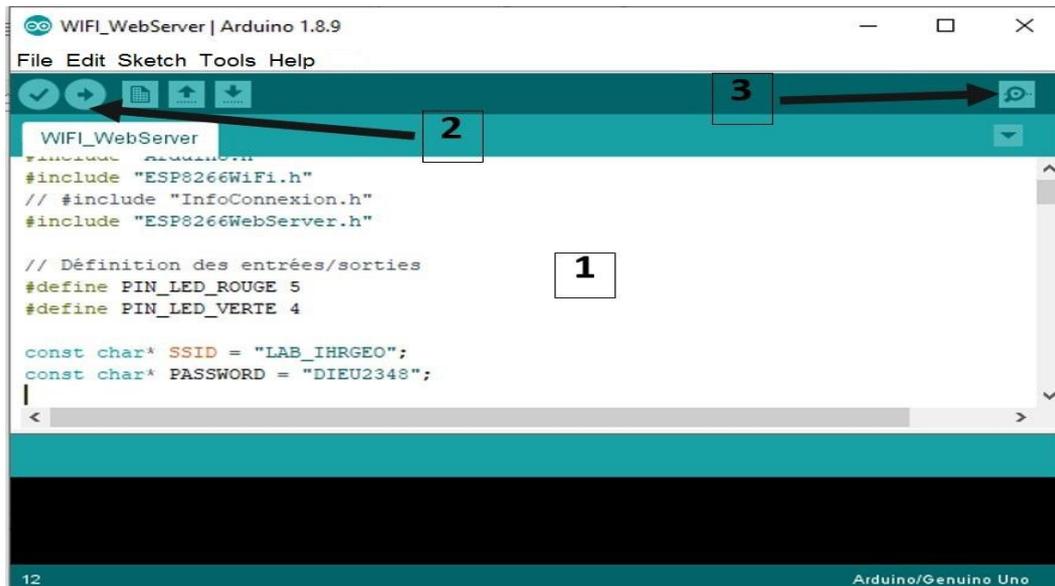


Fig. 4. Arduino Ide

2.1.5 LARAVEL FRAMEWORK

Laravel is an open-source web framework written in PHP that follows the model view controller (MVC) principle and is entirely developed using object-oriented programming (OOP). Laravel is distributed under the MIT license, with its source code hosted on GitHub. Laravel stands out as one of the best PHP frameworks for web application development. Designed to offer an elegant and expressive syntax, it provides a range of advanced tools and features that simplify the development process. It can handle the high level of complexity required for enterprise applications that require a backend, including user accounts, exports, and order management systems.

2.2 METHODS

The system (Fig. 5) consists of two parts: the software developed with the Laravel development platform and the hardware developed with Arduino Mega 2560 microcontrollers, SIM GPS / GSM SIM 808 and ESP 8266. In terms of the software, there are two types of interaction: interaction between the user and the system for modifying their information and that of the vehicles they own, and interaction between the administrator and the systems for performing all basic configurations. The hardware is organized around the Arduino MEGA2560 microcontroller, which enables real-time processing of the location data collected and is therefore of paramount importance. It ensures communication between the ESP 8266 module, the GPS / GSM SIM 808 module and the software (website). It receives geolocation data (longitude and latitude) and vehicle speed from the GPS / GSM SIM 808 module, then transmits this data via the ESP 8266 module to the website for processing and visualization of the vehicle's various positions on Google Maps. Figure 5 shows the general architecture of our system.

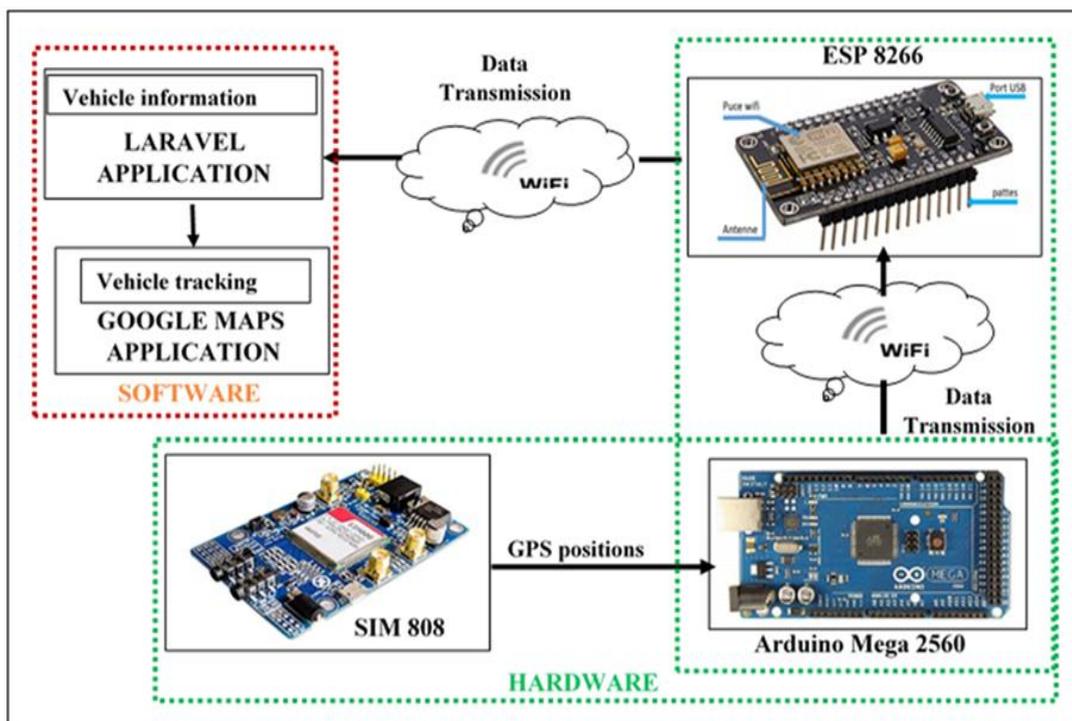


Fig. 5. Conceptual architecture of the system

2.2.1 SOFTWARE SECTION

The study is structured around the architecture below. There are two types of interaction: interaction between the user and the system for modifying user information and information about vehicles belonging to the user, and interaction between the administrator and the system for performing all basic configurations and tracking vehicle locations via the Google Maps platform.

- **Interaction between the user and the system**

This works as follows: the user logs in to access all the features available to them. After logging in, the user can modify their personal information, modify vehicle information if necessary, view the location of vehicles in real time, or display the history of the different positions taken by vehicles on a given day (Fig. 6).

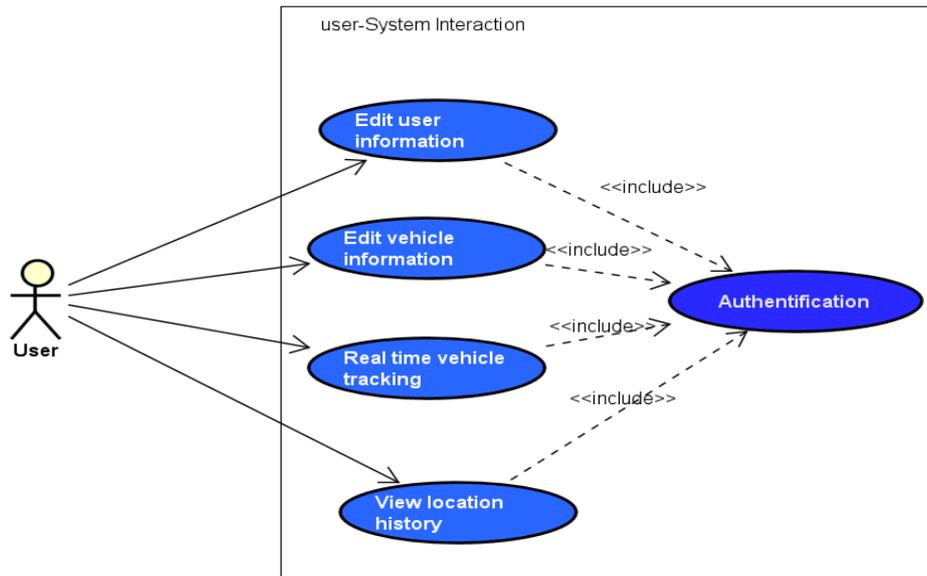


Fig. 6. Summarizes the interaction between the user and the system

• Interaction between the administrator and the systems

This represents the various interactions between the administrator and the system. Before performing any actions, the administrator must authenticate themselves and will then have access to all the actions available to the user mentioned in the previous section (recording of personal information). In addition to these actions, the administrator can add, modify, and delete users and vehicles. They can view all vehicles registered in the system on a map and access their entire history. (Fig. 7) summarizes the interaction between the administrator and the system (Fig. 7).

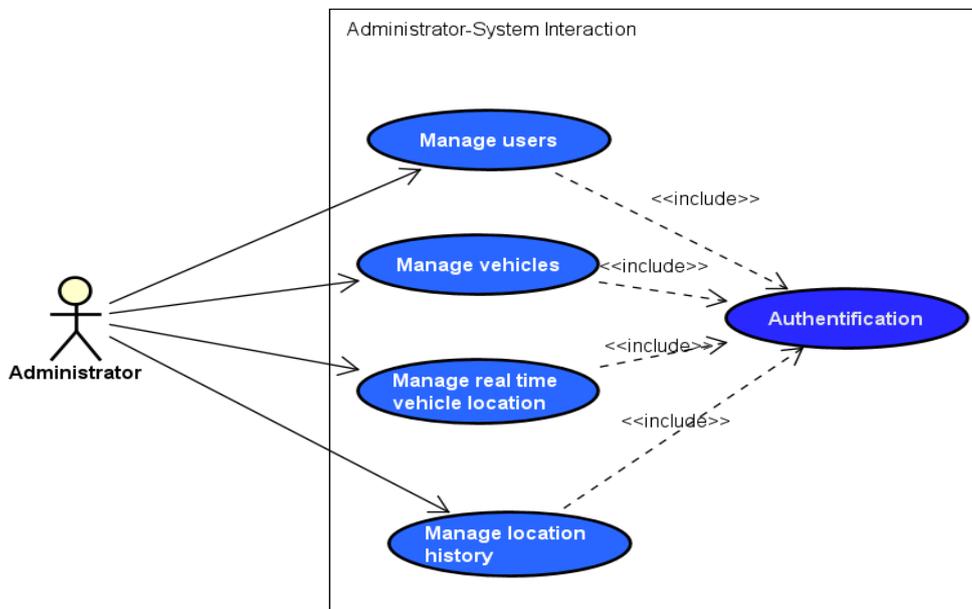


Fig. 7. Administrator-System Interaction

- **Google Map JavaScript API**

Users can view their vehicle and trip history for a given day in real time. This visualization is displayed on a map using the Google Map JavaScript API. The Google Maps JavaScript API (Application Programming Interface) is a library provided by Google that allows interactive maps to be integrated into websites. The Google Maps JavaScript API allows developers to add customized, interactive and dynamic maps to their websites. It provides a library of JavaScript features that allow maps to be manipulated in a variety of sophisticated ways. Its features include: Map display, it allows you to customize the appearance of the map (map type, style, zoom, etc.) and center it on a specific location. Adding markers: It allows you to mark points of interest on the map with custom icons. Road planning: The API can calculate the shortest roads between two points, taking into account different modes of transportation (car, walking, public transportation). Info windows, these allow additional information to be displayed when a user clicks on a marker. it offers roads by car, public transport, or on foot, get traffic information, view a place with Google Street View, find its location on the map by activating the smartphone's GPS and access satellite and 3D images.

2.2.2 HARDWARE SECTION

In this system, it should be noted that the Arduino board is the key component, as it establishes communication between the two modules: the GSM/GPS SIM 808 module and the ESP8266 module. The hardware was developed using the GSM/GPS SIM 808 module, which uses its GPS antenna to determine the vehicle's coordinates at a given time and then transmits them to the ESP8266 module, which establishes communication between the hardware (Arduino Mega 2560 microcontroller) of our system and the software (the website). Using the Arduino board, it transmits the vehicle's geographical coordinates to the software via an API for storage in a database. To ensure real-time tracking, the ESP8266 sends its actual position to the website every 30 seconds. It is also important to note that geolocation coordinates (latitude and longitude) can be obtained by sending an SMS. In fact, users who know the SIM card inserted in the GSM/GPS SIM 808 module can send a text message, which the module will read and then extract the number of the person who sent the text message to send them another text message with the longitude, latitude, and a Google Map link to view the position of their vehicle.

2.2.3 TEST DATA

To test our system, the information for ten (10) users will be recorded in Table 2. Similarly, vehicle information will also be listed in Table 3. The tables are presented below.

Table 2. User information table (All personal data presented are fictitious and used solely for system testing)

N°	First and last names	Email	Contact
1	AKA Jean Boris	ariane40107574@gmail.com	0758559152
2	OUATTARA Justin	justino@gmail.com	0504421091
3	AKA Jean Kevin	kevin225@gmail.com	0708688154
4	FOFANA Mohamed	tiebafof@gmail.com	0777856635
5	KOUADIO Nathan	knathan@gmail.com	0544782281
6	OWAGOKE Nathaniel	owagoke@gmail.com	0708084226
7	ASSEMIEN STEPHANE	stephass@gmail.com	0707287042
8	AHISSI JEAN REGIS	ahissi225@gmail.com	0777856635
9	BOTTIN Grace	gbottin@gmail.com	0546801383
10	DJIGUEMDE Esther	djiguemde01@gmail.com	0768306184

Table 3. Vehicle information table (All vehicle and contact information are fictitious and used for experimental validation only)

N°	Make	Model	Color	Registration	Phone number
1	SUZUKI	DZIRE	White	3929FL01	0173454545
2	HUNDAY	SONATA	Black	2012HV01	0504421091
3	Mazda	CX-5	Brown	224HU01	0708688154
4	BMW	X3	Black	1475HG01	0777856635
5	BMW	X5	Blue	AA-059-AA	0544782281
6	Hyundai	Accent	Brown	AA-118-GZ	0708084226
7	SUZUKI	S-PRESSO	Black	3926FL01	0707287042
8	HUNDAY	TUCSON	Black	1085HP01	0777856635
9	Mitsubishi	Mirage	Gray	861LS01	0789244243
10	KIA	SPORTAGE	Red	AA-195-GP	0170445746

3 RESULTS

3.1 GENERAL CONNECTION OF THE SYSTEM

The Arduino Mega 2560 board (Fig. 8.) is connected to the USB port of the laptop to be powered and receive the program. Then the GPS/GSM SIM 808 module is connected to a 9V power supply so that it can operate. We can also see that the ESP8266 module and the GPS/GSM SIM 808 module are connected to the Arduino Mega 2560 board, which establishes communication within the system. The GPS / GSM SIM 808 module determines the location of the vehicle and then sends it to the ESP 8266 module via the Arduino Mega 2560 board. This data is immediately sent by the ESP 8266 module over the local network to the computer to be used on the internet by our web application. This allows the user to access the location of their vehicle in real time on the Google Maps application.

- **Arduino-SIM808 Assembly**

The following diagram (Fig. 9.) represents a location system using the same GSM and GPS module integrated into the SIM808 module. This system is capable of determining the car’s position using a GPS receiver and also allows communication over cellular networks via a GSM mobile phone module which allows it to send its position information by SMS to the car owner. For this to work, it is essential that the module has a SIM card (telephone chip). All of this is programmed automatically by the Arduino mega 2560 board. This system uses SMS data transmission, the SIM808 module sets the coordinates (latitude and longitude) of the car’s current location. Once the GPS/GSM/GPRS module receives a call from a predefined number, it will reply with an SMS indicating the module’s GPS position. Based on its location with the mobile phone, the car’s location will be determined and sent to the user.

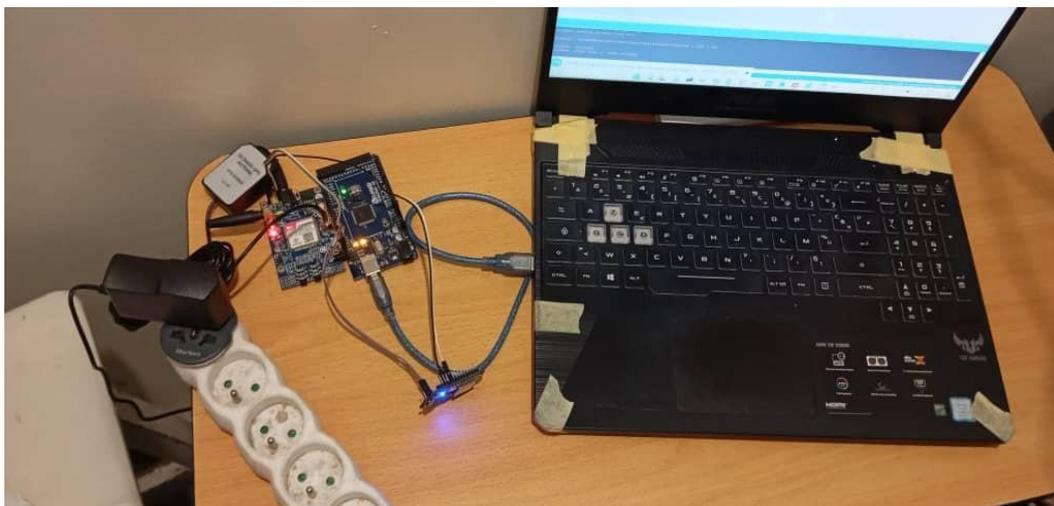


Fig. 8. General System connection

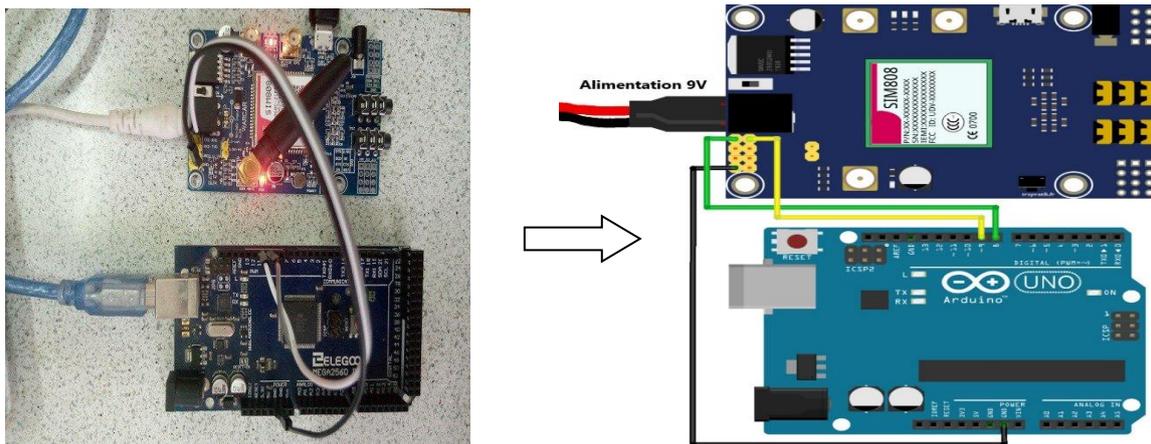


Fig. 9. Arduino-SIM808 Assembly

3.2 INTERACTION BETWEEN ADMINISTRATOR AND SYSTEM

When launching the application, the administrator must authenticate to access their space, which is why a login interface will be displayed (Fig. 10). In addition, if the administrator forgets their password, they can reset it by clicking on the “forgot password?” link on the login interface (Fig. 10). This will give them access to the password reset interface (Fig. 11). Once they have successfully completed the authentication process, the administrator will see their dashboard. On this dashboard, they can view all vehicles on the Google Map integrated into the system using red markers (Fig. 12).

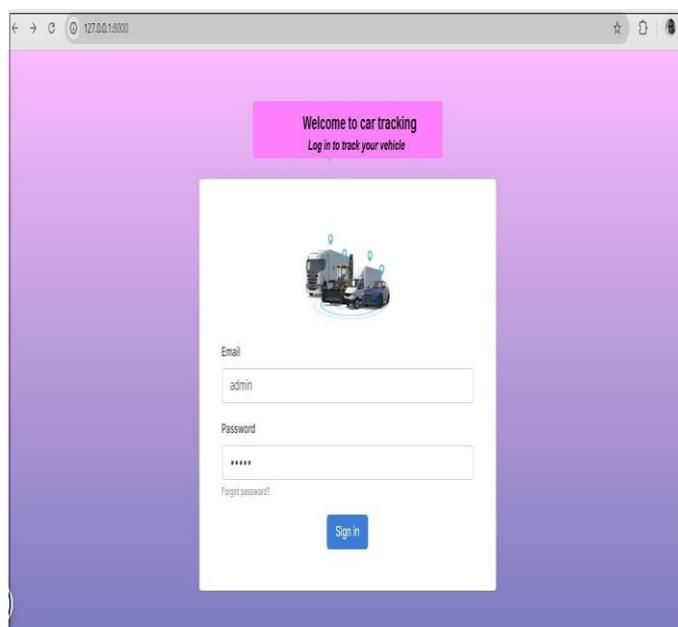


Fig. 10. Authentication Interface

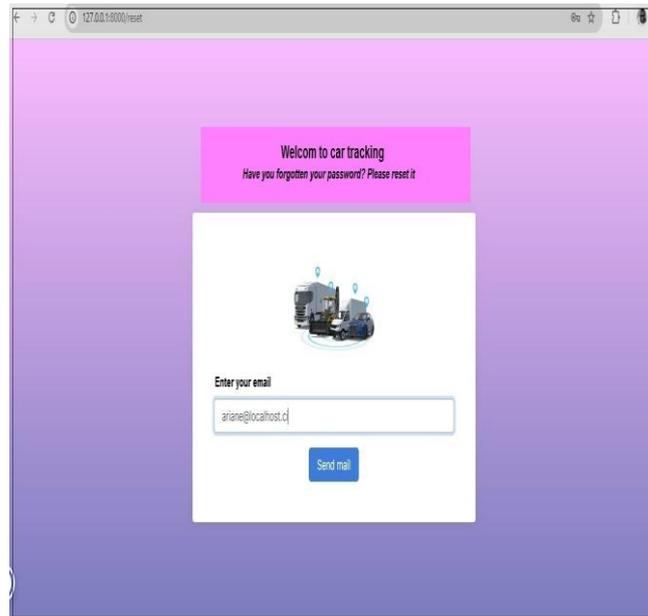


Fig. 11. Password Reset Interface

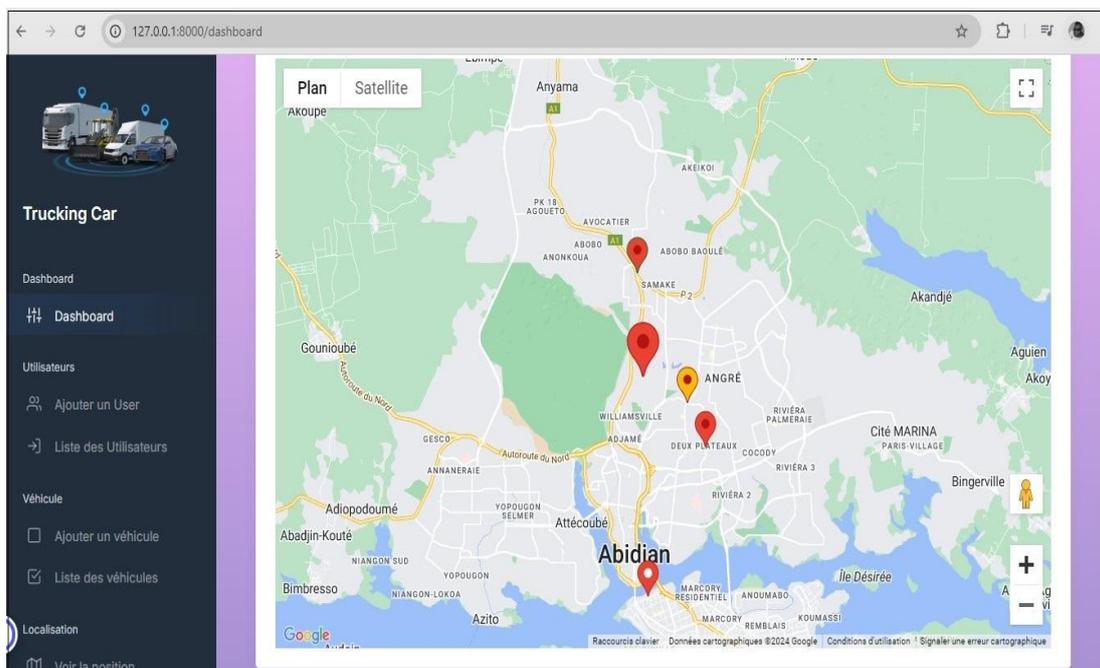


Fig. 12. Google Map Interface

3.3 USER MANAGEMENT

Clicking on Add User brings up the user add form (Fig. 13) with information such as surname, first names, email address, date of birth, etc. The administrator can also view all users registered in the application (Fig. 14). This is done by clicking on "User List" in the navigation pane. This interface allows you to manage users by deleting them if they decide to unsubscribe from our services, or by modifying their data in case of errors during registration.

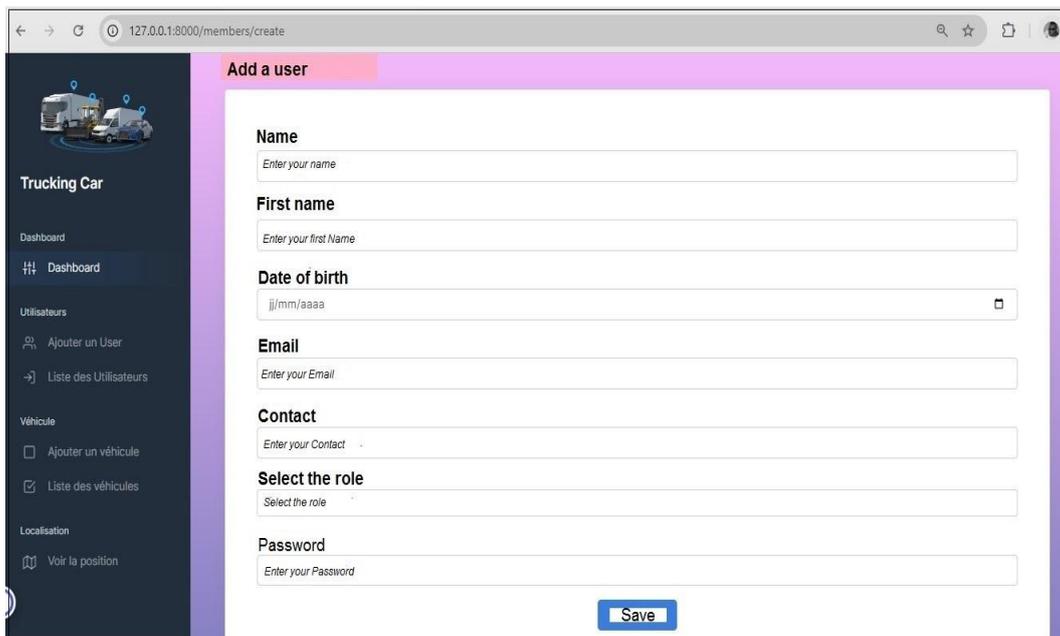


Fig. 13. User Add Interface

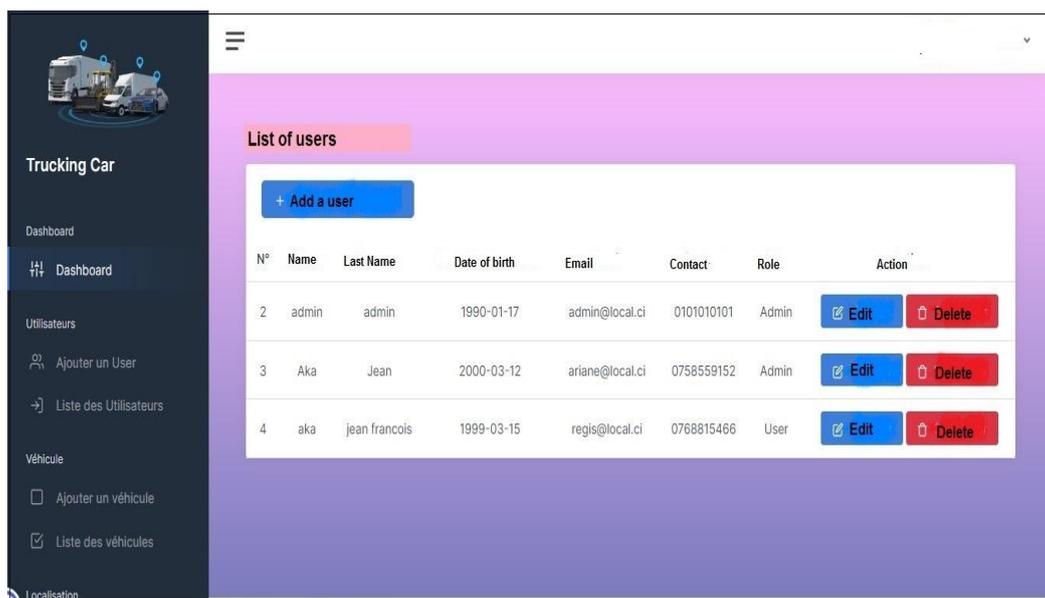


Fig. 14. User Viewing Interface

3.4 CAR MANAGEMENT

After user management comes car management, as the hardware device is installed in cars. To do this, a car addition form (Fig. 15) is used to enter the various vehicles, characterized by information such as make, registration number, color, etc., and to establish the user-vehicle correspondence. The administrator has access to an interface that allows them to view and manage vehicle lists (Fig. 16).

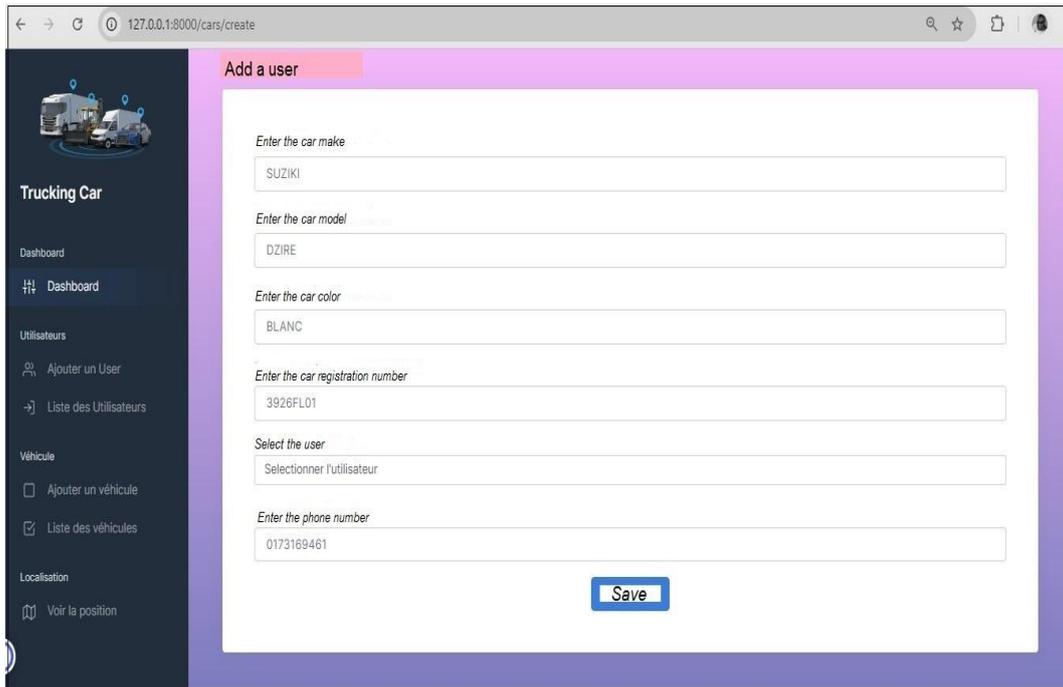


Fig. 15. Car Add Interface

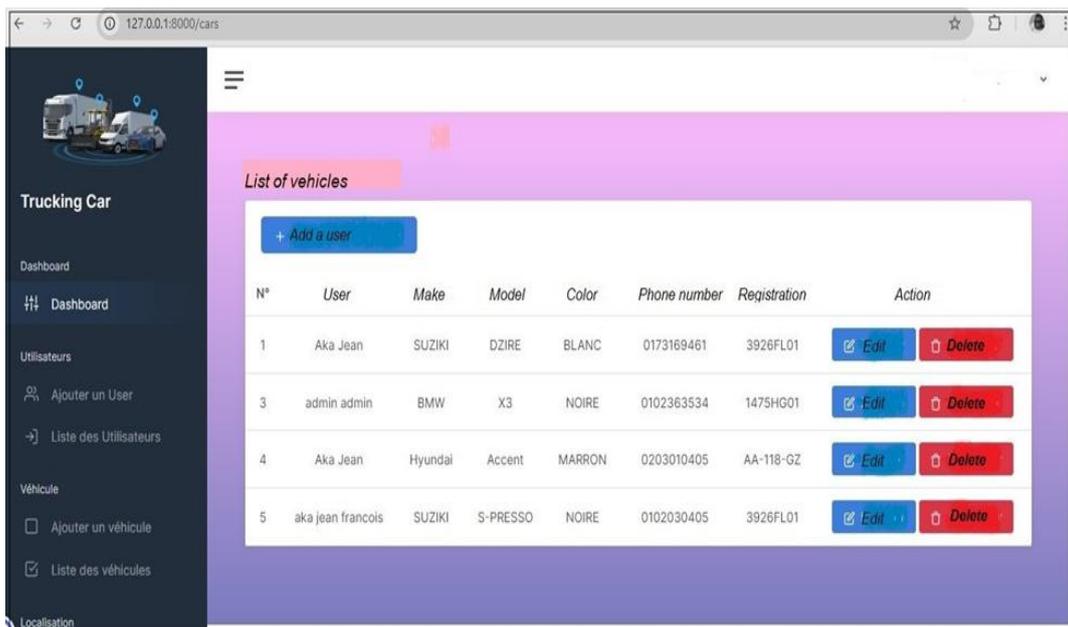


Fig. 16. Vehicle Viewing Interface

3.5 VEHICLE TRACKING

Finally, after completing the basic configurations, a table is available that allows you to view the real-time position of the vehicles and their positioning history on a Google Map integrated into our application (Fig. 17). By clicking on "display position," we obtain the real-time position of the 10 vehicles registered for testing our system (Fig. 18). By clicking on "position history," we obtain all the positions taken by a vehicle over a given period, then the distance traveled during that period and the time taken for the journey. Fig. 19 shows an example of a message received by the user when they request the location of their vehicle via a text message sent to the hardware system through the GSM/GPS SIM 808 module. Table 4 shows the different geographical coordinates and speeds of the 10 vehicles used to test our system. The SIM 808 GPS / GSM module with a GPS

sensitivity of -165 dBm has a good performance in challenging environments like urban or indoors places. This module enabled vehicle tracking with a speed accuracy (DGPS) of 0.05 m / s, an acceleration accuracy (DGPS) of 0.05 m / s² and a time accuracy of 10 ns, in this study test speeds ranging from 35 km / h to 65 km / h.



Fig. 17. Interface for viewing a vehicle's history on Map

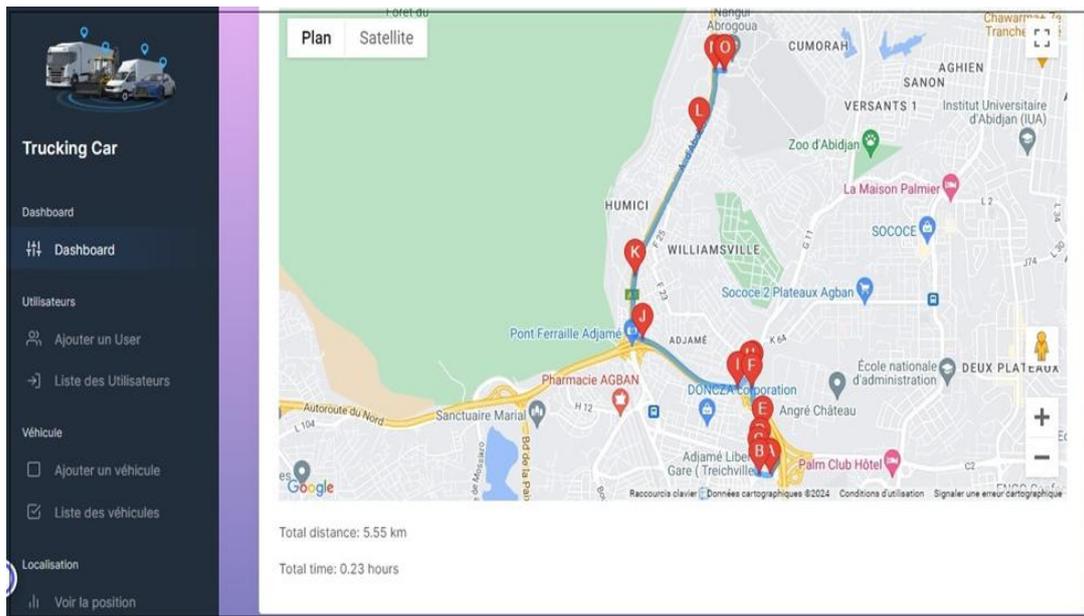


Fig. 18. Google map interface for tracking vehicle

Table 4. Geographical coordinates and vehicle speeds

N°	Latitude	Longitude	Speed
1	5.353457961261525	-4.0151636378049504	10 km/h
2	5.353502473487766	-4.01624753234791	24 km/h
3	5.354479957759769	-4.0163744872503	30 km/h
4	5.355194677278511	-4.016120011908582	30 km/h
5	5.357125543509917	-4.015864585256912	35 km/h
6	5.360834766740553	-4.017039452721567	36 km/h
7	5.361859758382305	-4.016988081998532	37 km/h
8	5.361840430660573	-4.01726063222412	38 km/h
9	5.361009397259376	-4.018545375957919	60 km/h
10	5.36513259247693	-4.028286738485645	65 km/h

Table 5. Geographical coordinate and vehicle distances

N°	Latitude	Longitude	Distance (m)
1	5.353457961261525	-4.0151636378049504	120,1
2	5.353502473487766	-4.01624753234791	
3	5.354479957759769	-4.0163744872503	109,6
4	5.355194677278511	-4.016120011908582	
5	5.357125543509917	-4.015864585256912	299,54
6	5.360834766740553	-4.017039452721567	
7	5.361859758382305	-4.016988081998532	540,9
8	5.361840430660573	-4.01726063222412	
9	5.361009397259376	-4.018545375957919	196,63
10	5.36513259247693	-4.028286738485645	

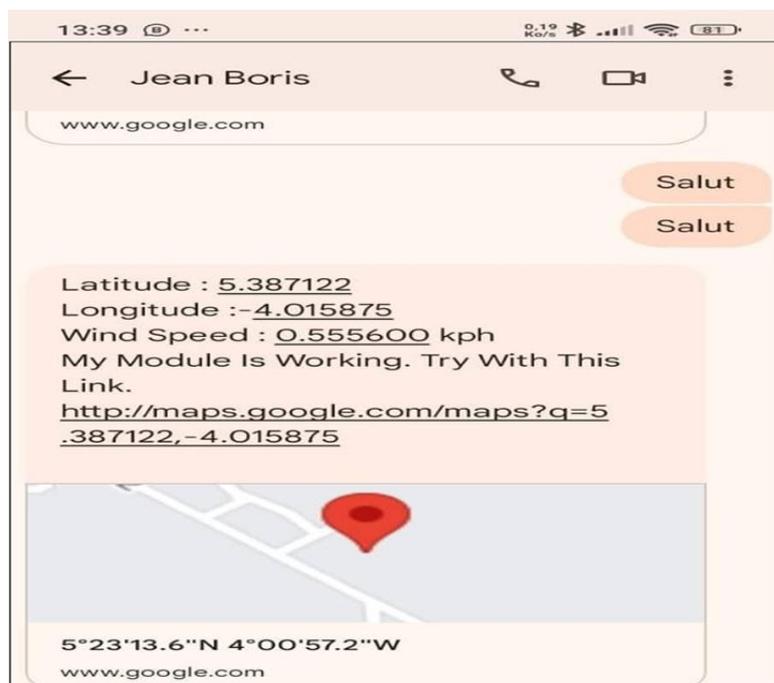


Fig. 19. GSM module message showing the geographical position of a vehicle

4 CONCLUSION

The objective of this project was to set up a system for geolocating vehicles. This system allows users to track their vehicles in real time. It consists of the following microcontrollers: Arduino Mega 2560, GPS/GSM SIM 808, and ESP 8266, which make up the hardware developed in the C++ programming language, which enabled the implementation of the location system and the sending of SMS messages via the GPS/GSM SIM 808 module and the transmission of data collected by the GPS/ GSM SIM808 to a web application that acts as the software component was provided by the ESP8266 microcontroller. This web application was developed in Laravel (a PHP framework) and JavaScript through the Google Map API (application programming interface) to exploit and display data on a map and calculate distances between two points. In addition, this study allowed us to use two (2) geolocation methods, namely real time location, which transmits geolocation data in real time to the database in order to save GPS coordinates and extract the route history and mileage for a given day, referred to as "tracking" in our work. And on demand location, which involves sending SMS messages to the system to obtain location data in rural areas where there is no network coverage and vehicle speed. In the future, this system will be printed on a PCB (printed circuit board) and integrated into a box with a rechargeable power source to ensure autonomy connected to the vehicle's battery in order to locate it even if the vehicle is turned off.

REFERENCES

- [1] D. Sperling and D. Gordon, « Two Billion Cars: Driving Towards Sustainability», In: Oxford University Press, 2009.
- [2] A. Altaf and N. Khan, « Auto-theft: Over 80 cars, bikes stolen in two months, Pakistan», In: The Express Tribune, 2016.
- [3] A. El-Rabbany, « An Autonomous GPS Carrier-Phased-Based System for Precision Navigation», Intelligent Transportation Systems Conference, 17-20 September 2006.
- [4] D.W. Kum, A. Khan and Y. Z. Cho, « Traffic Density-Based Broadcast Scheme for Vehicular Ad Hoc Networks», IEICE Transactions on Communications, Vol.E95-B, No.12, pp.3875-3878, Dec. 2012.
- [5] A. Khan, J. C. Nam and Y.Z. Cho, « ACK-Based Broadcast Protocol for Vehicular Ad Hoc Networks», IEICE Transactions on Communications, Vol.E97-B, No.5, pp.960-966, May. 2014.
- [6] M. S. Uddin, S. M. Allayear, N. C. Das and F. A. Talukder, « A location based time and attendance system», International journal of computer theory and engineering, Vol6, No1, pp 1-36, 2014.
- [7] S. W. Su and C. T. Lee, « Metal-frame GPS antenna for smartwatch applications», Progress in Electromagnetics Research Letters, 62, pp41-47, 2016.
- [8] R. C. Jisha, M. P. Mathews, S. P. Kini, V. Kumar, U. V. Harisankar, and M. Shilpa, « An android application for school bus tracking and student monitoring system», In IEEE International Conference on Computational Intelligence and Computing Research (ICIC), pp. 1-4, 2018.
- [9] J. Lluch, M. Rebollo, A. Calduch-Losa, and R. Mollá, « Precision of wearable GPS in marathon races», IEEE Consumer Electronics Magazine, 10 (1), pp.32-38, 2020.
- [10] J. Chung, Boyle J. and D. C. Wheeler, « Relationship Between Life-Space Mobility and Health Characteristics in Older Adults Using Global Positioning System Watches», Journal of Applied Gerontology, Vol 41, No4, pp.1186-1195, 2022.
- [11] Cullen, A., Mazhar, M. K. A., Smith, M. D., Lithander, F. E., Ó Breasail, M., & Henderson, E. J. (2022). « Wearable and Portable GPS Solutions for Monitoring Mobility», in Dementia: A Systematic Review. Sensors, Vol22, No9, pp33-36, 2022.
- [12] C. Shen, Y. Xiong, D. Zhao, C. Wang, H. Cao, X. Song and J. Liu, « Multi-rate strong tracking square-root cubature Kalman filter for MEMS-INS/GPS/polarization compass integrated navigation system», Mechanical Systems and Signal Processing, Vol 163, pp.108-146, 2022.
- [13] J. Jana, S. Basu, S. Tripathi, A. Chakrabarti and J. Bhaumik, (2022), « Design and Implementation of IoT-Based System for Tracking and Monitoring of Suspected COVID-19 Patient», In Advances in VLSI and Embedded Systems, pp. 163-173, 2022.
- [14] D. Patel, R. Seth and V. Mishra, « Real- Time Bus Tracking System», Int. Res. J. Eng. Technol., vol. 4, no. 3, pp. 743- 746, 2017.
- [15] M. N. Ramadan, M. A. Al- Khedher and S. A. Al-Kheder, « Intelligent Anti- Theft and Tracking System for Automobiles», Int. J. Mach. Learn. Comput., vol. 2, no. 1, pp. 83- 88, 2013.
- [16] H. Khaled, K. Tarek, W. Mohamed, M. Hussein, « Intelligent Transportation System Real-Time Tracking», Qeios, CC-BY 4.0, June 24, 2023.
- [17] S. Safdar, A. Zeb, A. Khan, Z. Kaleem, « Android based vehicle tracking system», AI Endorsed Transactions on Energy Weband Information Technology, pp.1 - 4, Volume 5, Issue17, 2018.
- [18] M. Drahanský, M. Charvát, I. Macek, J. Mohelníková (2013), « Thermal Imaging Detection System: A Case Study for Indoor Environments», Sensors 2023, 23, Vol7, pp.8-22, 2023.
- [19] B. Brik, N. Lagraa, N. Tamani, A. Lakas, Y. Ghamri-Doudane, « Renting out cloud services in mobile vehicular cloud», IEEE Trans. Veh. Technol., vol. 67, no. 10, pp. 9882- 9895, 2018.
- [20] Y. Izrailevsky, C. Bell (2018), « Cloud Reliability», IEEE Cloud Comput., vol. 5, no. 3, pp. 39- 44, 2018.