RESEARCH ARTICLE OPEN ACCESS

Embedded System Based Car Engine Locking and Safety System

S.Sona *, S.M.Sowmea *, V.Siva **
** Department of Aeronautical Engineering, Sri Ramakrishna Engineering College, Coimbatore- 641 022.*
*** Assistant Professor, Department of Aeronautical Engineering, Sri Ramakrishna Engineering College, Coimbatore- 641 022.*
Corresponding Author : S.Sona

ABSTRACT
The main objective of this application is to enhance the driver safety, by analyzing the road conditions and driver behavior by using sensors associated with android mobile. This system enhances the driver safety measures by identifying the driver behavior such as follows: (i) the driver wearing seatbelt or not (ii) Monitor the driving condition by using accelerometer connected in mobile phone and (iii) Locating the driver position by means of mobile GPS. This system allows further precedence only if the driver is wearing seat belt or else it produces the alert message like the driver is not wearing seat belt to administrator. In the proposed approach, once the driver entered successfully into the application, it shows the driver/vehicle position and in which direction the driver is proceeding on. As well as, this application is used to trap the location of the driver (vehicle) by using mobile Global Positioning System (GPS). For all the entire system is used to improve the safety measures of driver.

Keywords – Android mobile, Driver safety, Global Positioning System, Seatbelt, Vehicle.

Date Of Submission: 21-06-2019 Date Of Acceptance: 05-07-2019

I. INTRODUCTION

Intelligent transportation systems (ITS) introduce advanced applications aimed at providing innovative services, offering traffic management and enabling users to be better informed, including support for safety, mobility, and environmental applications. In parallel to ITS, mobile devices have experienced technological breakthroughs in recent years, evolving towards high performance terminals with multi-core microprocessors. The smart phone is a clear representative outcome of this trend In addition, the on board diagnostics (OBD- II) standard, available since 1994, has recently become an enabling technology for in-vehicle applications due to the availability of Bluetooth OBD-II connectors. These connectors enable a transparent connectivity between the mobile device and the vehicle's electronic control unit (ECU). When combining high performance smart phones with OBDII connectivity, new and exciting research challenges emerge, promoting the symbiosis between vehicles and mobile devices, and thereby achieving novel intelligent systems. Driving Styles implements a solution based on neural networks, which is capable of characterizing the driving style of each user, as well as the fuel consumption. In order to achieve this functionality, the data is obtained from the ECU via the OBD-II Bluetooth interface, including the speed, acceleration, revolutions per minute of the engine, mass flow sensor (MAF), manifold absolute pressure (MAP), and intake air temperature (AIT). Currently, this information can be collected and used in applications aimed at improving road safety and promoting eco-driving, thus reducing fuel consumption and greenhouse gas emissions. Specifically we find that, by shifting towards a more efficient driving style, users can save up to 20% of fuel while improving driving safety, thereby reducing greenhouse gases as we detail later on.

Our proposed architecture applies data mining techniques to generate a classification of the driving styles of users based on the analysis of their mobility traces. Such classification is generated taking into consideration the characteristics of each route, such as whether it is urban, suburban, or a highway, and it is then correlated with the fuel consumption and emissions of each driver. To achieve the overall objective, our system comprises four elements:

1) An application for Android, based smart phones. Using an OBD-II Bluetooth interface, the application collects control information (by default every second, but it is configurable by the user) such as speed, acceleration, engine revolutions per minute, throttle position, and the vehicle's geographic position. In addition, we also obtain via OBD-II the mass flow sensor (MAF), the manifold absolute pressure (MAP), and the intake air temperature (AIT) that are used in the calculation of fuel consumption. After gathering the information, the user can upload the collected data to the remote data center for analysis.

2) A data center offering a web interface to collect large data sets sent by different users concurrently and to graphically display a summary of the most relevant results, like driving styles and route characterization of each route sent. Our solution is based on open source software tools such as Apache, PHP and Joomla.

3) A neural network which has been trained using the most representative route traces in order to correctly identify, for each path segment, the driving style of the driver, as well as the segment profile: Urban, suburban or highway. We use the back propagation algorithm, which has proven to provide good results in classification problems such as the one associated to this project .

4) Integration of the tuned neural networks both within the mobile device itself, and in the data-center platform The goal is to use neural networks to dynamically and automatically analyze user data, reporting to the drivers in real time and allowing them to find out their driver profile, thus promoting a less aggressive and more ecological driving.

The first step for a user is to register at http://www.drivingstyles.info, and to download the free Android application. After installing the Android application in the mobile device, and after connecting to the Bluetooth ELM327 interface inside the car (this connector is mandatory on all vehicles since 2001), the data acquisition process will start. The Android application is a key element of our system, proving connectivity to the vehicle and to the Driving Styles web platform. Currently, it can be downloaded for free from the Driving Styles website1, or from Google Play 2. Once the mobile application is installed and configured, the user must pair the mobile device with the ELM327 (OBD-II Bluetooth device) to start getting data. The data obtained from the different variables such as acceleration, engine revolutions per minute (RPM), speed, mass flow sensor (MAF), manifold absolute pressure (MAP), and intake air temperature (IAT) are analyzed by the application, showing users the characteristics related to their driving, fuel consumption, and CO2 emissions. In order to adjust the application functionality, it offers several configuration options, i.e., user creation, connection options, GPS activation, sensor sampling, and type of fuel definition. The second main component of our architecture corresponds to the data center and its web interface. To implement this component, we have selected open source software such as Apache HTTP, and Joomla as the content management system (CMS). We have used a CMS, combined with the use of a resource wrapper, which detach our system from the presentation layer, thus focusing on the driving styles characterization problem.

1.1 Existing System

In past systems, the driver cannot be get controlled by any activities, only they can be controlled by means of traffic constables and other respective individuals. This causes the road accidents to be increased day-by-day. For this occurrence a new system is required to solve the issues caused because of failed to wear the seat belts while driving. The major reasons of accidents include violation of traffic rules, drunk driving, carelessness etc. Lots of developments have shown that most of the severe injuries and death case can be prevented to more than 50 percent if a person wears a seat belt and avoids driving without that with have a disadvantages of Possibility of Road Accidents is more, Failed to detect the location of riders, Analyzing the speed level and position is difficult, Not proper and accurate outcomes.

1.2 Proposed System

In the proposed approach of smart road safety measures lots useful features are provided to drivers to avoid road accidents and provides a safe riding on roads. . Initially it asks the driver to register their details into server. In the proposed approach, the system Safety Checking process means wearing the seat belt or not, which is done by using Proximity sensor. Once the Seat belt is weared properly, then only the application allows the driver to proceed further. This application provides provision to identify the driver's (or) Vehicle Location instantly as well as summarize the Vehicle Movement Status to alert and avoid road Accidents with a advantages of Road Accidents are highly avoided, Location is properly detected using mobile GPS, Analyzing the speed level and position with proper norms, 100% accurate and proper outcomes are grasped by means of GPS and Proximity sensor.

II. MODULE DESCRIPTION

2.1 Driver Registration Process

The driver details registration module allows the new drivers to register his/her details into our android application, which has lots of identity attributes such as Driver Name, Mobile Number, Address and so on. This is an entry point to all drivers to register the identity and which it leads to login into the application.

2.2 Checking For Safety

The safety checking module is an important feature of the application, which is triggered out by the proximity sensor presented into the android mobile. The application triggers the mobile application to use the proximity sensor to verify the driver is wearing the seatbelt or not.

2.3 Vehicle Location Tracking

The vehicle location tracking module is efficiently triggered by means of mobile GPS and which returns the longitude and latitude details of the driver to identify the location of the respective driver. As well the vehicle movement detection option is helpful to finding out the present position of the vehicle such as either forward, backward, left or right.

2.4 Alert for Accidents

The alert module is helpful to identify the situation of driver this module triggers the mobile GSM to send the SMS to the respective person number and indicates the critical situation of the respective driver.

III. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

3.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

3.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

3.3. White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

3.4 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. Cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

3.5 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

IV. OPEN DATABASE CONNECTIVITY

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change. Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN. The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn’t as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

4.1 JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of “plug- in” database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on. To gain a wider acceptance of JDBC, Sun based JDBC’s framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution. JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after. The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively.

V. FIGURES AND TABLES

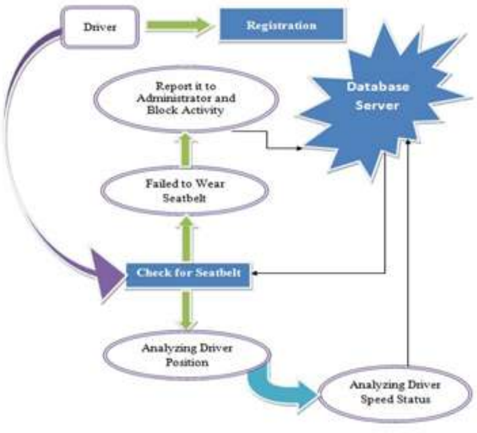
Driver
Registration
Report it to Administrator and Block Activity
Database Server
Failed to Wear Seatbelt
Check for Seatbelt
Analyzing Driver Position
Analyzing Driver Speed Status

Figure 1. System Architectural Design.

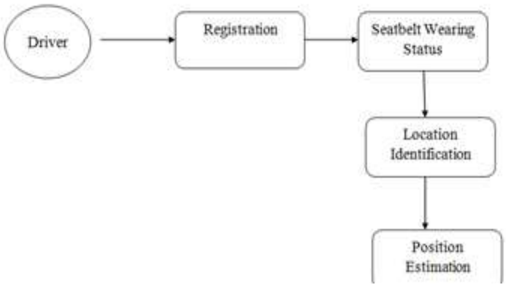

Figure 2. Data Flow Diagram.

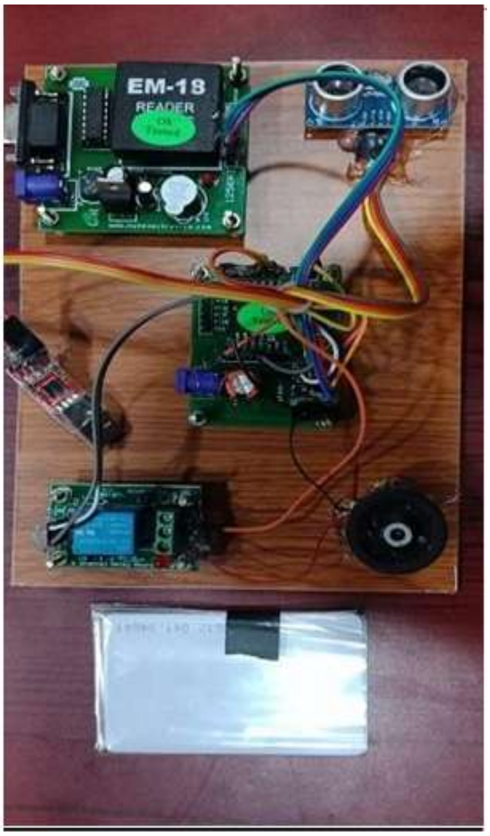

Figure 3. Implementation of Model

System requirements both the software requirements and hardware requirements are presented in the following table.

Description	Quantity
System	Pentium IV 2.4 GHz
Hard Disk	40 GB
Floppy Drive	1.44 Mb
Monitor	15 VGA Colour
Mobile Device	Android Smartphone

Table 1. Hardware Requirements

Description	Quantity
Operating system	Android
Technology Used	Kit Kat
Coding Language	Java

Table 2. Software Requirements

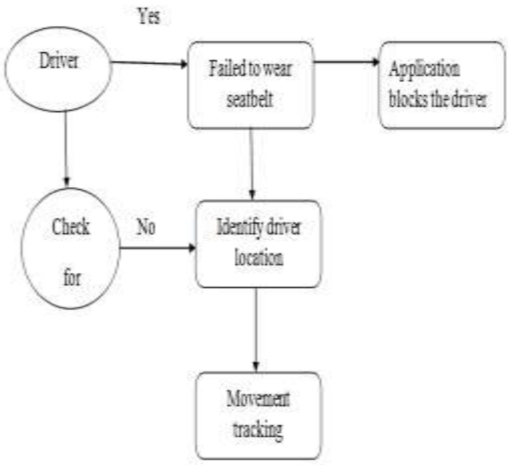

Figure 4. Data Flow Diagram.

VI. CONCLUSION

From the above work, we can concluded the following aspects as a result of a present work.

1) Intelligent vehicle and driver safety system ensures the security of the vehicle which is a main issue in today's life also the security of the vehicle as the number of accidents of the drivers has a great number. .
2) To start the vehicle it will make necessary for the driver to wear the seatbelt which is a very important substituent for the driver safety. This application will also help in tracking the vehicle if it gets stolen or dragged somehow by the thief by simply sending an SMS to the owner about the present location of the vehicle and the status regarding the position that the driver drives the vehicle.
3) In future the proposed work is further extended to provide some useful features such as route recommendations based on real-time feedback about the congestion state of different alternative routes, as well as providing estimated greenhouse emissions for different routes. Enable the GPS location identification system to monitor the driver's present location via GPS map facilities and provides guidance to drivers in case of any emergency scenarios

ACKNOWLEDGEMENT

We express my heartfelt thanks to our esteemed Managing Trustee and Joint Managing Trustee for their kind earnest attempt to provide all the facilities and interest they showed in the welfare of students as well as faculty community.

We thank our Principal Dr.N.R.Alamelu, for her benevolent patronage in all the way to carry out this work successfully.

REFERENCES

[1]. A.Meintz, and M. Ferdowsi "Designing efficient hybrid electric vehicles," IEEE Veh. Technol. Mag., vol. 4, no. 2, June 2009.

[2]. U. Hernandez, A. Perallos, N. Sainz, and I. Angulo "Vehicle on board platform: Communications test and prototyping," in Proc. IV, 2010, pp. 967−972.

[3]. H. Johansson, P. Gustafsson, M. Henke, and M. Rosen gren "Impact of EcoDriving on emissions," in Proc. TAP, Avignon, France, 2003.

[4]. D. Y. C. Leung, D. J. Williams, "Modelling of Motor Vehicle Fuel Consumption and Emissions Using a Power-Based Model," Environmental Monitoring and Assessment, vol. 65, pp. 21–29, 2000.

[5]. M. Andr´e "Driving cycles development: Characterization of the methods," SAE Technical Papers Series 961112, 1996 .

S.Sona" Embedded System Based Car Engine Locking and Safety System" International Journal of Engineering Research and Applications (IJERA), Vol. 09, No.06, 2019, pp. 31-35