Prashant Shah

by Prashant Shah Prashant Shah

Submission date: 08-Jul-2020 11:26AM (UTC+0530)
Submission ID: 1354891431
File name: Final_Year_Project_-_Report_-_PLAGARISM_CHECK.docx (2.19M)
Word count: 5056
Character count: 28229

TITLE: CAR FLEET MANAGEMENT

GUIDE NAME: Dr. Venkatesh K, Professor, Department of Computer Science & Engineering

Authors:

1. Prashant Shah 1NT16CS214
2. Ankit Prasad Gupta 1NT16CS216
3. Bivek Kumar Shah 1NT16CS171

CHAPTER 1: INTRODUCTION

1.1. Background

Fleet Management is the process in which fleet and asset information is managed in proper way by which it enables the company to reduce the cost, improve efficiency and ensure compliance across an entire fleet operation. [6]

Car Fleet Management (CFM) allows any organization like Zoom Car, Ola, Uber to monitor the condition of the vehicle and manage it in a proper way. Like the existing Truck Fleet Management system this will provide beneficial result to the car leasing organizations. It also has some additional features relevant to car fleet management.

1.2. Brief history of Technology/Concept

Rapid advance and adoption of technologies associated with Internet of Things has resulted in the availability of sophisticated miniaturized sensors and processing elements available for the Automotive platform also. The CFM system built using the IOT platform. In addition, some of the features like information on driving habits of specific drivers, eye blinking patterns among others can benefit from the use of Machine Learning algorithms to detect unusual behavior patterns and provide warnings for suitable corrective measures to be implemented.

1.3. Applications

1. Safe driving: The IOT vehicle platform of CFM can help the drivers and the organization to maintain a safe driving record. The IOT platform provides collision avoidance systems that use sensors to detect imminent collision and provides warning to the driver, it alerts the driver if he/she is driving over the speed limits, and it also has ultrasonic sensors which are used to determine presence of objects in close proximity which will aid in giving guidance during parking of the vehicle. A camera module is also integrated to the platform. This module will aid in determining if the driver is sleepy. If he is found to wink at a very low frequency (when compared to his normal range of winking frequency) then it can be safely inferred that the Driver is either sleeping/sleepy and appropriate alert message can be sent to him.

.

2. Connected World: With the increasing use of the services of IoT, the world will be connected in future. Every person will be able to communicate remotely with

everything and everyone. This system will provide, the connected service to the organization and the driver driving the vehicle. This will help in real time monitoring of the vehicle by the Fleet manager.

1.4. Research motivation and Problem statement

1.4.1. Research Motivation

Car lease company like Zoom, Ola Rental, Uber are facing a lot of challenges in order to track and know the condition of their vehicle. They are also busy maintaining the record and authorizing the driver. By seeing these difficulties and with the rapidly growing technology Internet of things (IOT). We proposed to the solution to automate the manual process and to use the truck fleet management concept to the car fleet management. By this it can help the company to manage their car fleet and keep it safe from the fraudulent drivers.

1.4.2. Statement of the Problem

There are a lot of things that can go wrong with vehicle such as Driver Making unwanted stops, driving recklessly, not taking the best routes, or wasting energy. That were not able to be tracked using legacy systems and technology can now be harnessed to track these. The IoT platform will have sensors for:

- Positioning sensing
- Geofencing
- Vehicle Power control (On/Off)
- Drowsiness Detection
- Ultrasonic Parking
- Speed Monitoring
- Automatic Wiper Control

1.5. Research Objectives and Contributions

1.5.1. Primary Objective

Car Fleet Management is basically developed with the help of Internet of Things and Machine Learning which helps the organizations giving car for a lease to the public. It is very hard to manually monitor the fleet of vehicles in real time and advise from the central office to forecast and prevent accidents, damage to the vehicle, theft of vehicles from the parking lots, theft of parts of the vehicle etc. So, this project could be very

helpful for those organization to remotely monitor and control the vehicle so that accidents can be avoided, detect damage to vehicle parts, theft of parts, etc. Hence, the main objective of this project is to mitigate all these problems mentioned and help the organization like Zoom Car, Ola, Uber, etc. by automating the vehicle and provide facility for remote status monitoring and control.

1.5.2. Main Contributions

Car Fleet Management is the system developed for the organization like Zoom Car, Ola, Uber, etc. This system is developed using mainly two technologies, Internet of Things (IoT) and Machine Learning (ML). Hardware consists of different sensors and microprocessors and IoT Gateway. Programming Languages: Python and Embedded C are used in microprocessor for handling the low-level hardware. Similarly, the Dashboard application software at the central office consists of a backend developed using the Firebase Real time database and JavaScript, and the front end is developed using HTML, CSS, JavaScript, Bootstrap.

1.6. Organization of The Report

The rest of the report is organized in the following manner: Chapter 2: Literature Survey, discussing various methods presented by researchers of their proposed system. Chapter 3: System Requirements specifications. Chapter 4: Design. Chapter 5: Implementation of the project. Chapter 6: Test Cases. Chapter 7: Result. Chapter 8: Impact of the project towards society/environment. Chapter 9: Conclusion. Chapter 10: References.

1.7. Summary

The main aim of this project is to help organization facing different problem like theft of car parts, unusual behavior of customer (driver), going beyond the speed limit etc. Also, it will help the public to get such great service and enjoy the best feature which can satisfy their needs and help them not to worry much and leave every responsibility to this system. This system when implemented in its entirety can provide a lot of benefits to car leasing organizations. The system is being implemented in a modular manner so that new features can be easily incorporated as the need arises. .

CHAPTER 2: LITERATURE SURVEY

2.1. Introduction

Car Fleet Management is one of the IoT based project which allows organization like Zoom car, Ola, Uber to monitor their leased vehicle in proper way. Fleet Management System include different parameters like GPS tracking, Geofencing, Fuel level indicator, speed monitoring, etc. As an existing system, paper [1] basically deals only with Intelligent Transportation System (ITS) which provide some features like GPS tracking, NFC payment, etc. on transportation vehicle. Whereas paper [2] is totally based on fleet management system which provide only fuel level monitoring and GPS tracking. Apart from this paper [3] has introduced some new features of GUI interface and Linux based embedded microprocessor. Paper [4] just talked about the drowsiness detection where author have proposed one algorithm which alert driver when driver is about to sleep. Paper [5] is basically based on road accident. It provides the method how the road accident can be decreased.

2.2. Related work

The paper [1] described Intelligent Transportation System (ITS) consists of three components; the sensor system, monitoring system and the display system. Using GPS, NFC, Temperature and Humidity sensor they take every sensors data. Such sensors are connected to the server through IP address and port number. It uses HTTP protocol for request and response. With the successful connection of servers, databases (location DB, Commuter DB, Ambience DB) accept the value from the sensors. The webpage is designed with PHP which receive information from sensor and redirects the values to their respective server database through GSM module. Information Processing Systems (IPS) converts raw data (sensor data) into context data (meaningful useful data) and provide information to the bus driver.

The paper [2] explained the methodology of monitoring the fuel level and correlate it with the distance by the vehicle. This can also help in stopping pilferage of fuel. Here, firstly fuel level sensor and GPS based odometer sends the information to the controller. Then, the received data will be converted into transmittable form. Such data will be again sent to the Control database which will be operated with internet in a suitable and

convenient medium. Finally, after complete processing the status of the vehicle can be visible onto the Webpage of this application.

.

The paper [3] proposed "The Real Time Vehicle Fleet Management and Security System" project which is built on a Linux based embedded microprocessor. It uses GPS for vehicle location tracking. GSM-GPRS modem is used for communication whereas for security purpose physical panic button, Biometric sensor, Camera, and speakers are used. "In-Vehicle System" uses embedded Linux cubie truck whose main purpose is to get all the vehicle details using different sensors. "Server System" accept the data from In-Vehicle System and maintain a database to store the accepted data and the GUI server for user interface. For User Interface a dedicated server used for data acquisition and a GUI renderer is created. This GUI-renderer plots and displays the real time data dynamically.

The paper [4] uses Artificial Intelligence-based advanced algorithms to detect driver fatigue and the rate at which the driver is drowsy. It proposed an algorithm that uses eye and mouth vertical distances, eye closure, yawning and other engineered facial features to detect driver drowsiness.

The paper [5] proposed a system to prevent road accidents and to sense speed of vehicles during road travel and also to transmit data to the cloud. It uses the concept of WSN and IOT. It uses a sim module to transmit the collected information to the cloud. It uses crash sensor to detect the crash and if any emergency occurs it alert the nearby police station and hospital.

2.3. Study of Tools/Technology

The main two technologies used in Car Fleet Management are Internet of Things (IoT) and Machine Learning (ML). Every IoT project including Car Feet Management have three main systems: Sensor system, Monitoring System and Display system. Sensor system uses different types sensor where sensors are connected to microcontroller. Different microcontroller uses different tools. For example, Node MCU and Arduino mainly uses Embedded C, Raspberry Pi uses Python or Java. The data obtained by the sensor is now sent to the Monitoring system. Here, monitoring system helps to monitor

and control every data and process as required. Different databases are required to store the data. Driver Drowsiness Detection can be implemented using various methods; Machine Learning, MATLAB, etc. Finally, after every processing of the data, the required result is displayed in the GUI based Display system. Display system can be developed using different platform. Some of the examples are, MEAN Stack, HTML CSS, JavaScript, Android Studio (only for android application), ThingWorx, ThingSpeak, etc.

2.4. Summary

Car Feet Management provides a solution to a car lease company by providing best effective result in which company can monitor and control the status of Car in real time. The existing fleet management consists of some parameters like fuel monitoring, speed calculation, distance calculation, drowsiness detection with separate projects. However, from the above study it is clear that there doesn't exist a unified solution for all the requirements listed earlier.

CHAPTER 3: SYSTEM REQUIREMENTS SPECIFICATIONS

3.1. General Description

3.1.1. Product Perspective

Car Fleet Management consists of one dashboard for administrator where he/she can monitor the status of the car. It requires active internet connection for the dashboard because the system is built on the basis of Internet of Things and there are different sensors which will be sending the data to the administrator. The different sensors mentioned below are utilized to perform the required functions outlined in the proposal. CMS relies heavily on the wireless data network. As the CMS is designed for car leasing companies of the type outlined earlier, and their area of operation is mostly in metropolitan regions this key requirement of data connectivity is expected to be satisfied without any reservations.

3.2. Hardware Requirements

1. Must have an electric vehicle where CMS can be implemented.
2. Must have microcontrollers to process every elements of CMS.
3. Must have a sensor to monitor car's location (latitude & longitude).
4. Must have four sensors to monitor objects in the close proximity within a range of two meters.
5. Must have a sensor to monitor the speed of vehicle.
6. Must have a sensor to monitor the power supply on/off.
7. Must have a camera module to monitor driver's eyes activity for drowsiness detection.
8. Must have a sensor to operate automatic wiper when it starts raining.
9. Must have a Buck Converter to step down the DC voltage from the input to the output.
10. Must have a Display Screen for driver to view all the parameters.

Hardware Modules: A prototype model is built in order to demonstrate the features of the application. The hardware items used are:

- Electric Vehicle (Auto–Rickshaw)
- Raspberry pi
- Node MCU
- Arduino
- Buck Converter

- Display Screen
- Sensors (GPS, Ultrasonic, Accelerometer, Relay, Camera Module, Servo)

3.3. Software Requirements

We will be requiring the following software

3.3.1. Functional Requirements

1. Must have a valid updated Raspbian OS installed on Raspberry pi to process every element of CFM
2. Must have an Arduino software installed in system to process microcontroller.
3. Must require a proper Full Stack processing environment to create front end and back end of the software of CFM.
4. Must require a valid Firebase to store, process and evaluate the data.

Software Building blocks/tools: The software building blocks or tools used are:

- Raspbian OS
- Arduino Software
- Full Stack
- Firebase Real Time Database

3.3.2. Non-functional Requirements

1. Maintainability: This system requires good maintainability care from the organizations. Since, at a time there might be high number of drivers taking car on lease, it definitely requires good maintainability services to maintain the administration dashboard and monitor vehicle.
2. Reliability: Car Fleet Management must be reliable without any faults and bugs. Every mentioned parameters of the Car Fleet Management should be mentioned and monitored properly in the dashboard.
3. Scalability: The Firebase real time database must be scalable to adopt high number of drivers' detail. Since this application requires login usage to be stored in the database, there should be no cases of inconsistency in the scalability of the database.
4. Security: Since, Car Fleet Management is mainly dependent on Google API and Firebase Real Time database, there is high chance of the data loss due to hackers and attackers. So, the system must be secured by implementing different services

like, installing SSL, using anti-malware software, regular backup etc. This is the future implementation of this project.

3.4. Summary

If this system is maintained efficiently, the organization can get good benefits from it. Every hardware and software should be maintained up to date. If any replacement of the sensors and other hardware is required then it should be replaced in time. Workers working in the organization must take care of the proper working of the sensors. This system requires proper maintenance and care from the organization.

CHAPTER 4: DESIGN

4.1. Architectural Design

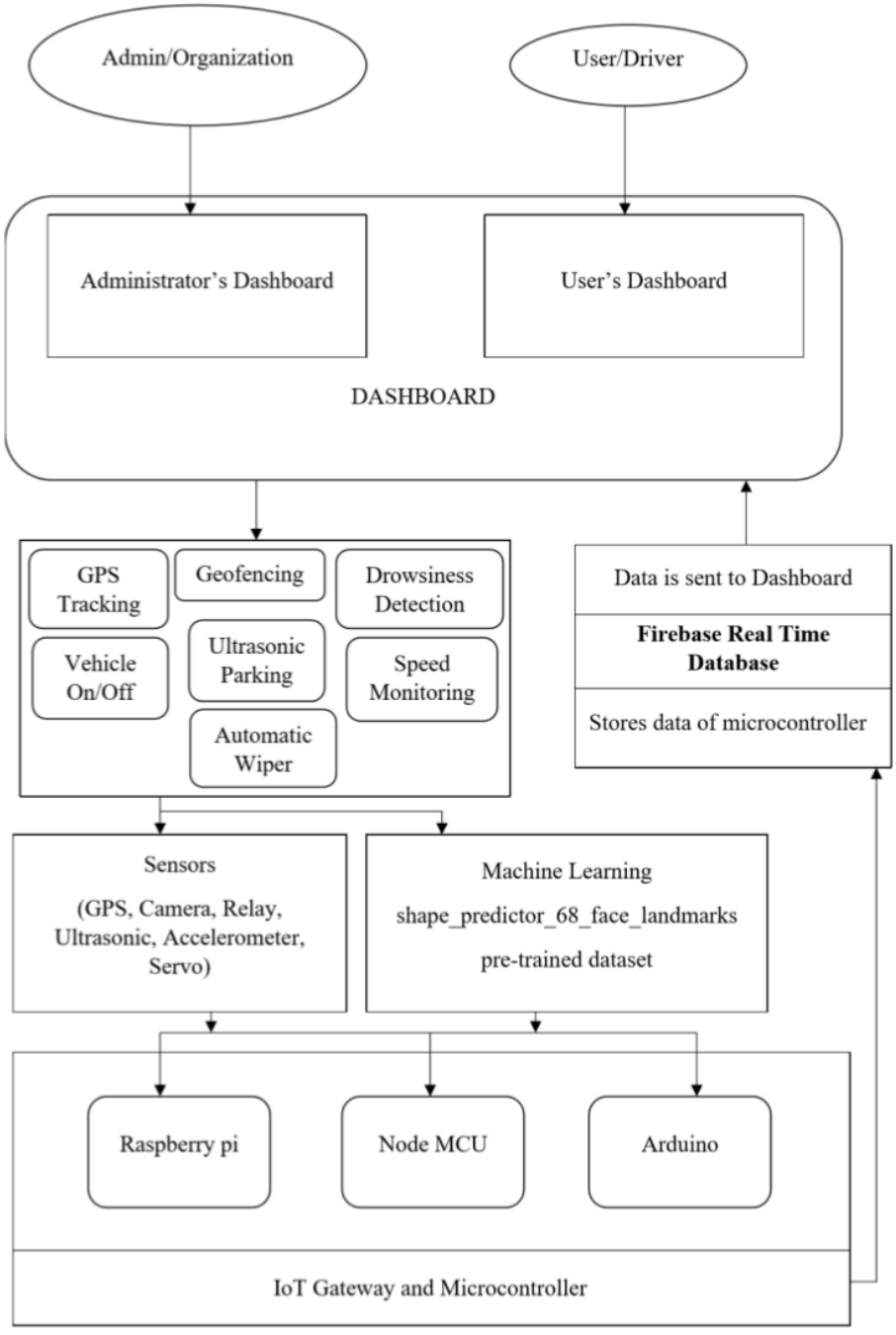

Fig 1: Architectural Design

4.2. Dataflow Diagram

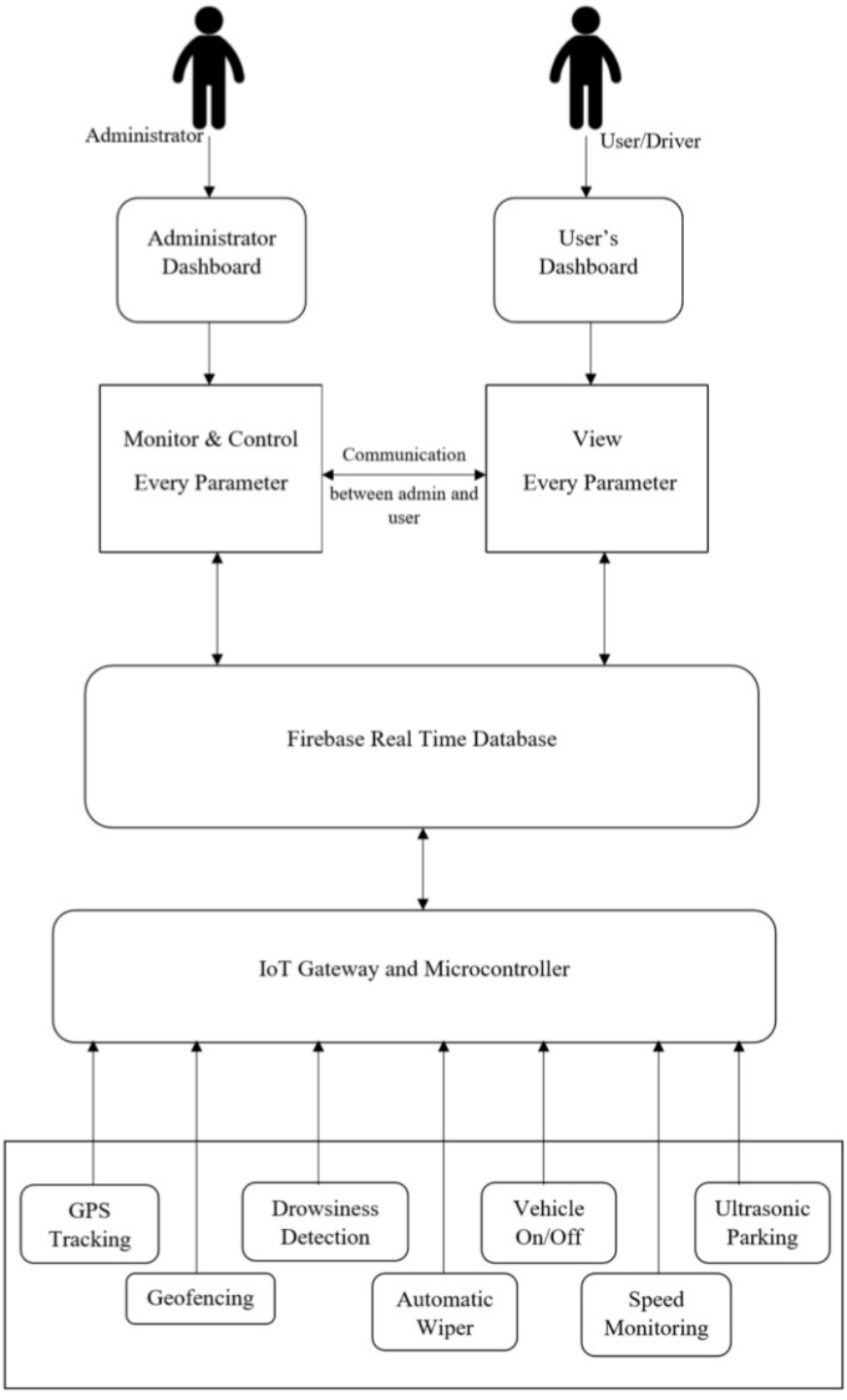

Fig 2: Dataflow Diagram

4.3. Class Hierarchy Diagram

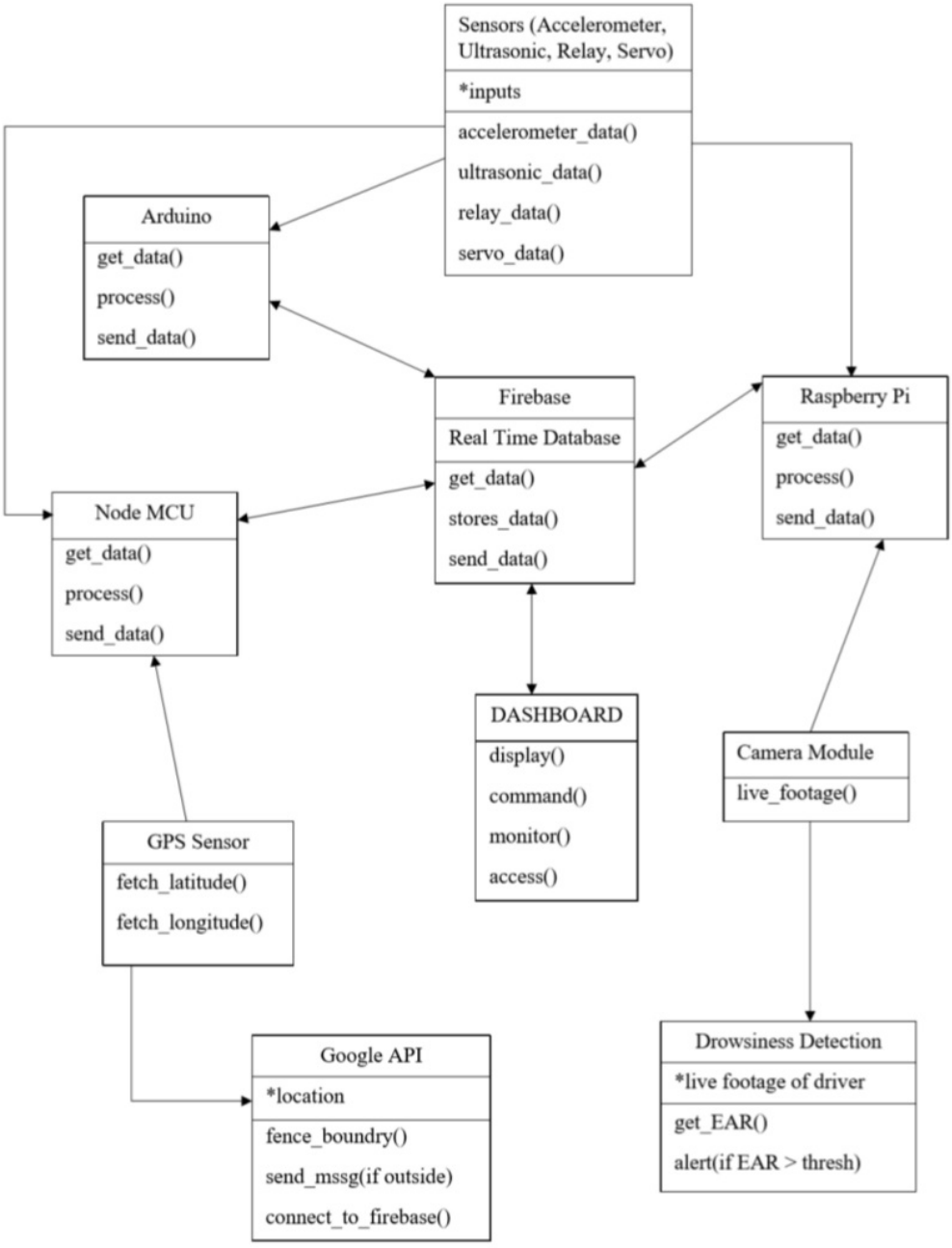

Fig 3: Class Hierarchy Diagram

4.4. Use Case Diagram

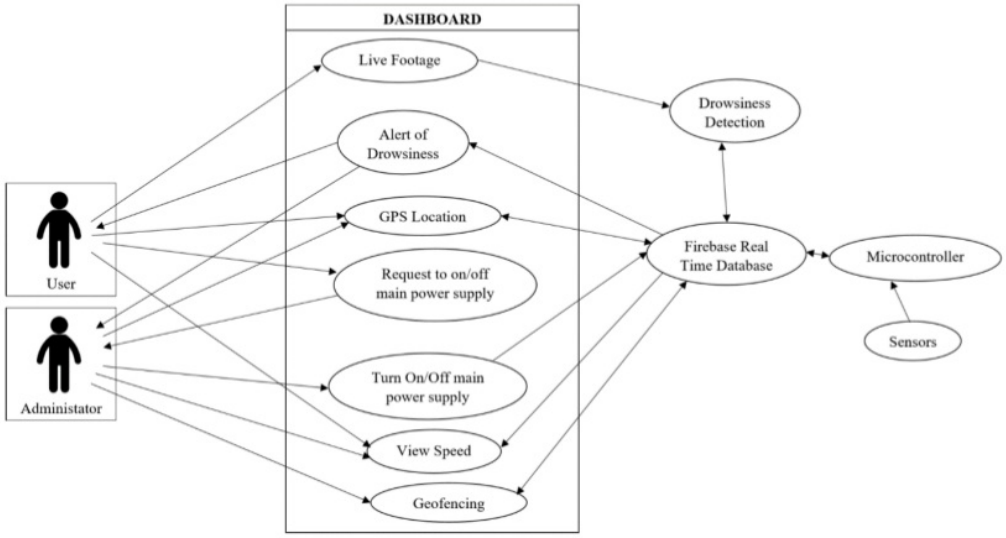

Fig 4: Use Case Diagram

4.5. Sequence Diagram

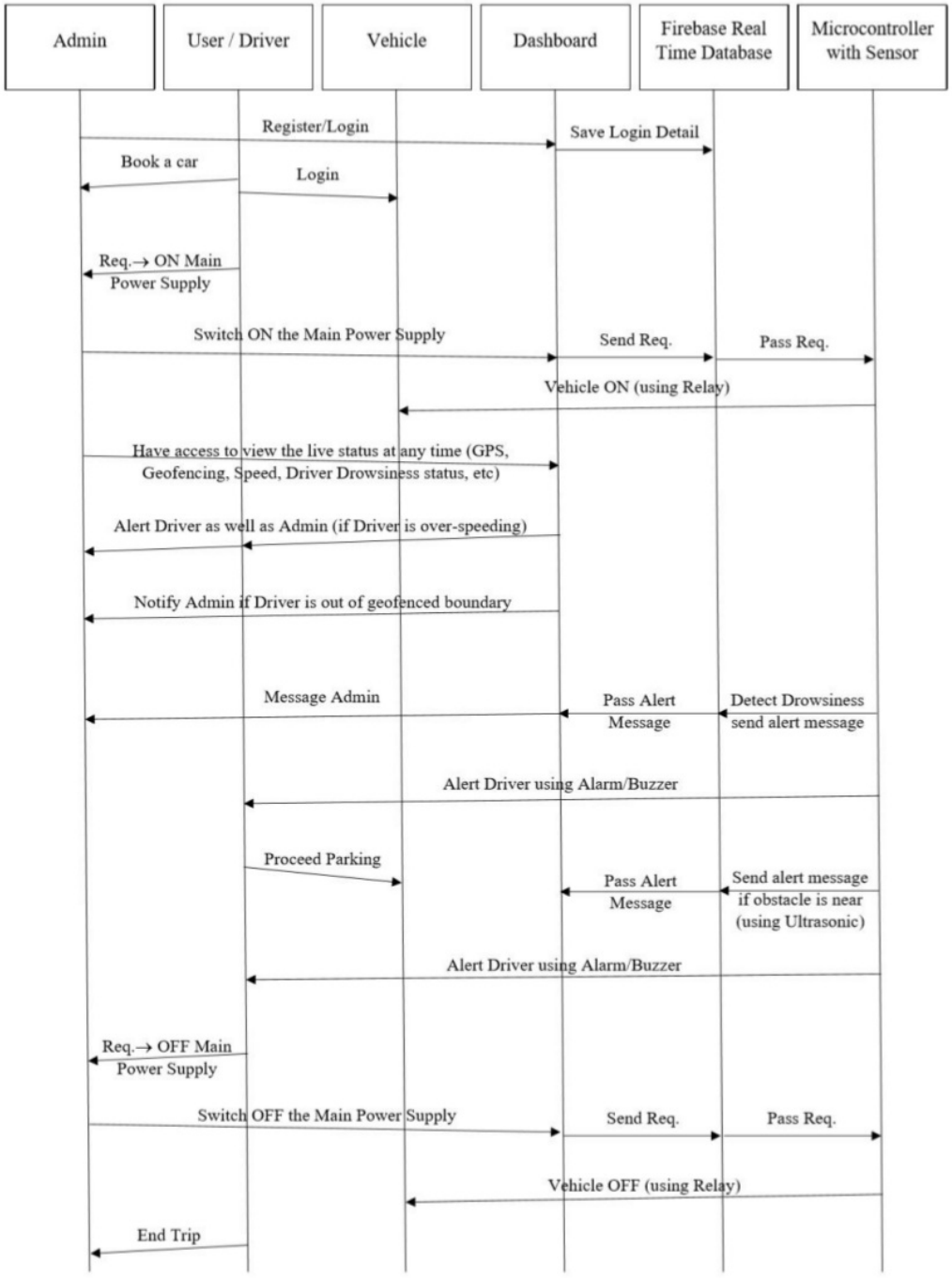

Fig 5: Sequence Diagram

4.6. Activity Diagram

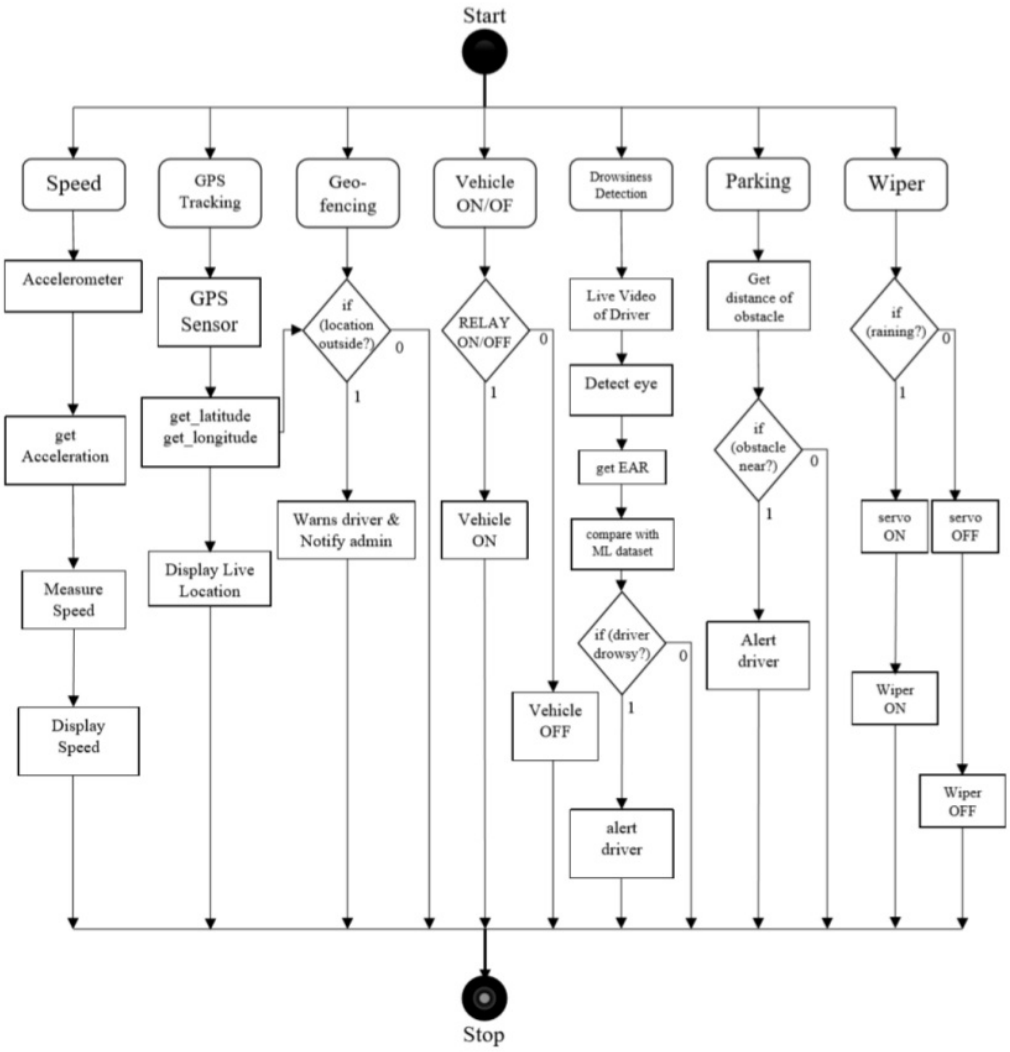

Fig 6: Activity Diagram

CHAPTER 5: IMPLEMENTATION

5.1. Methodology

System consists of seven modules: GPS Tracking; Geofencing; Vehicle On/Off; Drowsiness Detection; Ultrasonic Parking; Speed Monitoring; Automatic Wiper. Here administrator have one dashboard where he/she can monitor the vehicle which is taken for the lease.

All the data of the sensors are stored in the firebase real time database. The data is fetched from firebase using

5.2. Description of Process

The system is prepared for the company like Zoom Car, Ola, Uber, etc. which provides services of car to public as a lease. And with this there is high chance of damage of the vehicle part in a developing country like India. So, this system based of Internet of Things and Machine Learning is prepared to help those company to monitor all the vehicle taken on lease.

Entire Fleet Management consists of seven modules. Each modules' process are explained below:

a) GPS Tracking:

Live location of the driver can be visible in the dashboard using GPS Tracking. This system uses Node MCU and GPS module to fetch the live location of the vehicle taken on lease and displays to administrator he/she can easily monitor the live location of the vehicle in the real time basis.

Fig 7: GPS Tracking Example

b) Geofencing

Geofencing uses GPS to define the geographical boundaries. It allows administrator to set up triggers for the vehicle so when vehicle enters/exits the boundaries an alert message is generated and sent to the administrator.

Fig 8: Geofencing Boundaries in Google Map

This system uses Google Map API to fence the geographical boundaries and show it on dashboard. When the GPS location of the vehicle goes outside the boundaries then it notifies the administrator.

c) Vehicle On/Off

This system allows administrator to turn on/off the main power supply of an electric vehicle remotely. If the vehicle is parked in any remote location, then there is high chance of the vehicle being stolen. So, the control of the main power supply of an electrical vehicle can be handled directly by the administrator.

Relay sensor is used to switch on/off the main power supply. Remotely turning on/off doesn't mean that without administrator driver can't stop the vehicle. Starting/Stopping of the vehicle is fully done by driver. But just the main power supply which provide electricity to the vehicle is monitored by the administrator. Here before driver starts the vehicle, he/she needs to take permission from the administrator to turn on the main power supply. And after turning on the main power supply driver can start the vehicle. After ending the trip driver needs to inform the administrator the end of the trip and administrator will turn off the main power supply.

Fig 9: Example of Remote Vehicle

d) Drowsiness Detection

There might be situation where driver can get sleepy during the ride which can lead to accident. Such drowsiness can be detected using this system. This system will detect if the driver closes eyes for more than five seconds.

This system uses Machine Learning to fetch the EAR (Eye Accept Ratio) and alert driver if he/she closes eyes for more than 5 seconds. It will be annoying if the alarm keeps buzzing every time the driver blinks the eye.

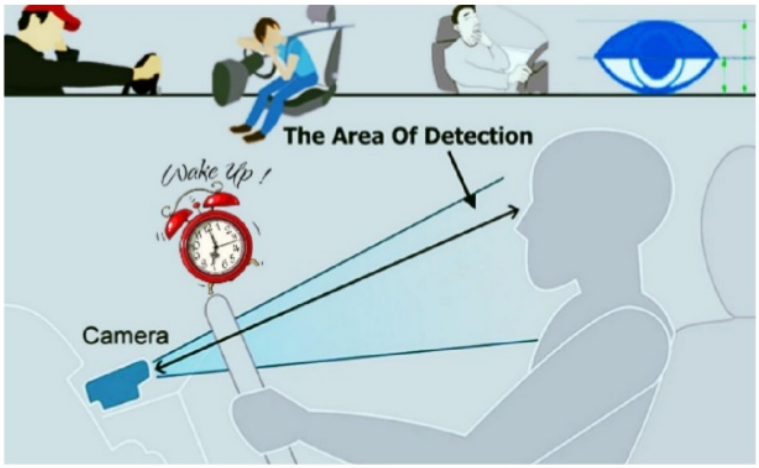

Fig 10: Drowsiness Detecting

ML uses shape_predictor_68_face_landmarks dataset and understands the difference between the closing of eye and opening of eye and alert an alarm to the driver if and only if he/she closes eyes for more than 5 seconds. The live footage of the driver is taken by camera module. The input live video is read as the frame using OpenCV and detects eye in every frame of the video. It checks if the closing eye is detected for more than 5 seconds continuously then alarm will be buzzed.

e) Ultrasonic Parking

It is sometimes very difficult to view all objects behind the car using only the rearview mirror while parking. So, Ultrasonic Parking can be very useful in such cases. This system uses Ultrasonic sensor which works on the principle of reflection of wave. Ultrasonic wave is transmitted by sensor which travels in air and when it gets obstructed by any material it gets reflected. The reflected wave is captured by the ultrasonic receiver module. Distance can be calculated by using the formula:

Distance = Speed × Time [7]

If distance is less than a prefixed threshold then the driver will be alerted that there is a likelihood of collision with an obstacle. This will help in preventing damage to the vehicle while parking.

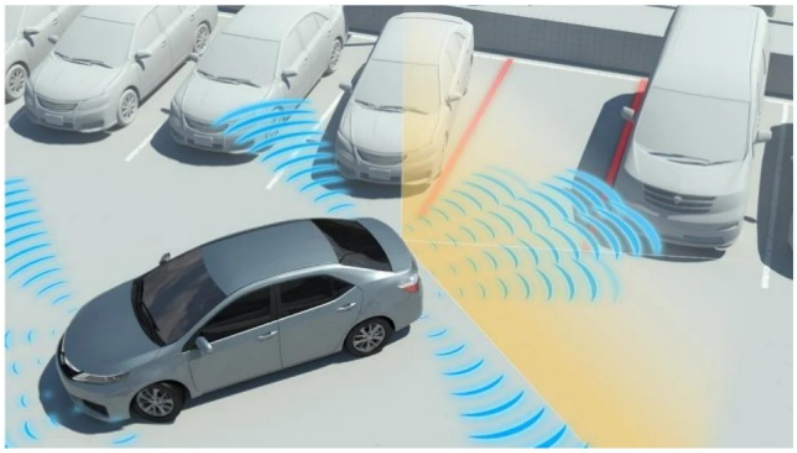

Fig 11: Ultrasonic Parking Example

f) Speed Monitoring

Most of the accident nowadays occur due to over speed in India. Car Fleet Management allows the organization to monitor the current location as well as the speed of the vehicle on the administrator's dashboard. If the speed of the car crosses the predefined limit then the driver will be alerted by the administrator. This can be implemented using accelerometer sensor. Using Raspberry pi and accelerometer sensor we can calculate the speed of the vehicle and display the speed on the dashboard of administrator.

Fig 12: Picture of Vehicle Speedometer

g) Automatic Wiper

Automatic Wiper can be one of the best examples of automatic vehicle component. This system helps driver not to worry much about the wiper switch. Whenever rain droplets fall on the rain sensor, servo motor will start rotating with an angle of 90°. By this the wiper will also turn on automatically.

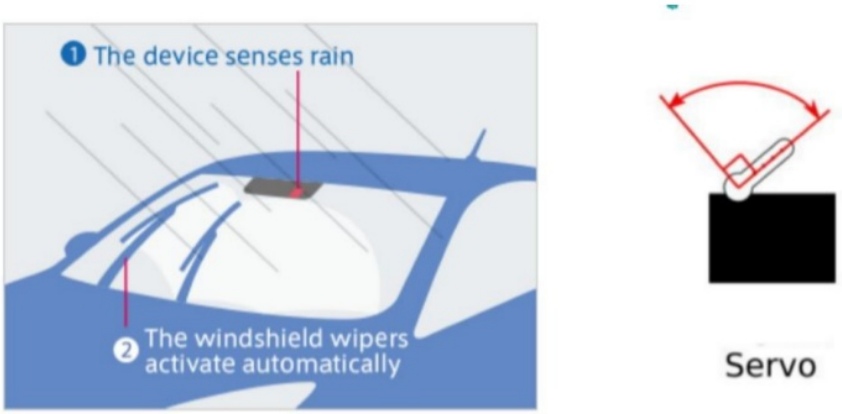

Fig 13: Automatic Wiper and Servo

5.3. Pseudo-code

Car Fleet Management consists of seven modules. Each modules' pseudo code are as follows:

i) GPS Tracking

GPS Tracking is maintained by GPS module and microcontroller. Pseudo-code for GPS Tracking with Node MCU is below:

```
WiFiServer server(80);
void setup()
{
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
void loop()
{
  while (ss.available() > 0)
    if (gps.encode(ss.read()))
    {
if (gps.location.isValid())
{
latitude = gps.location.lat();
lat_str = String(latitude , 6);
Firebase.pushString("Latitude", lat_str); //setup path and send readings
      Serial.println(lat_str);
      longitude = gps.location.lng();
      lng_str = String(longitude , 6);
      Firebase.pushString("Longitude", lng_str); //setup path and send readings
      Serial.println(lng_str);
    }
}
```

ii) Geofencing

Geofencing is obtained using the GPS location obtained from above method passing to the below geofencing html code. With the help of google map api it will help the system to maintain the vehicle to restrict inside the dedicated fenced area only.

Body:

```
  <div id="googleMap" style="width:100%;height:400px;"></div>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAdkVWN0Lkgo5mYllPfPtWnfp5EE6d3W8U&callback=myMap">
</script>
Starting Location (lat, lon):
      <span id="startLat">???</span>&deg;, <span id="startLon">???</span>&deg;
Current Location (lat, lon):
      <span id ="currentLat"> locating... </span> &deg;, <span id ="currentLon"> locating... </span>&deg;
Distance from starting location:<br/>
      <span id="distance">0</span> km
      Are we here?
      <span id="message">detecting....</span>
      Something New (new):<br/>
      <span id="new">locating...</span>&deg;
   <script type="text/javascript" src="https://code.jquery.com/jquery-2.1.4.min.js"></script>
   <script type="text/javascript" src="fence.js"></script>
   <div id="map" style="width:100%;height:500px"></div>
   <script>
function myMap1() {
//confidential code for setting up map
  myCity.setMap(map);
}
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAdkVWN0Lkgo5mYllPfPtWnfp5EE6d3W8U&callback=myMap1"></script>
```

iii) Vehicle On/Off

Vehicle can be turned on/off remotely using Arduino and relay. The pseudo-code is given below:

```
void setup() {
  char input;
  pinMode(relay,OUTPUT);
}
void loop() {
  if(Serial.available()>0) input = Serial.read();
  if(input == '1') digitalWrite(relay,HIGH);  //1 => Turn ON
  if(input == '0') digitalWrite(realy,LOW);  //0 => Turn OFF
}
```

iv) Drowsiness Detection

Drowsiness Detection is done using Machine Learning with python code. The pseudo-code is given below:

```
while True:
    ret, frame = cap.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    subjects = detect(gray, 0)
    for subject in subjects:
        shape = predict(gray, subject)
        shape = face_utils.shape_to_np(shape)  # converting to NumPy Array
        leftEye = shape[lStart:lEnd]
        rightEye = shape[rStart:rEnd]
        leftEAR = eye_aspect_ratio(leftEye)
        rightEAR = eye_aspect_ratio(rightEye)
        ear = (leftEAR + rightEAR) / 2.0
        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)
        cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)
        if ear < thresh:
```

```
        flag += 1
        print(flag)
        if flag >= frame_check:
           cv2.putText(frame, "****************ALERT!****************", (10,
30),
                       cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
           cv2.putText(frame, "****************ALERT!****************", (10,
325),
```

v) Ultrasonic Parking

Ultrasonic Parking is maintained using Raspberry pi and ultrasonic sensor. The python pseudo-code for the execution to calculate the distance of obstacle and alert the driver is given below:

```
GPIO.setwarnings(False)
      def buzzer():
      GPIO.output(Buzzer, GPIO.HIGH)
      def get_distance():
      GPIO.output(TRIG, True)
      time.sleep(0.00001)
      GPIO.output(TRIG, False)
      while GPIO.input(ECHO) == False: start = time.time()
      while GPIO.input(ECHO) == True: end = time.time()
      signal_time = end-start
      distance = signal_time / 0.000058
      return distance
      while True:
      distance = get_distance()
      if distance < 25:       buzzer ()
```

vi) Speed Monitoring

Speed of the vehicle is monitored by using Accelerometer and Arduino. The pseudo-code is as following:

```
void setup()
 {
```

```
 Xrest=analogRead(accelerometer_pin0);
 }
void loop()
{
  Serial.print("Time ");
  t1=millis();
  Serial.println(t1*0.001);
  Xread = analogRead(accelerometer_pin0)-Xrest;
  Gx=Xread/67.584; //Gx => Acceleration
  Serial.print("Speed:");
  Serial.print(Gx*t1*0.001);
  delay(700);
 }
```

vii) Automatic Wiper

Automatic Wiper is developed using raspberry pi with python code. The pseudo-code is given below:

```
p = GPIO.PWM(servoPIN, 50)
p.start(2.5)
def wiper(iterations):
    for x in range(iterations):
        p.ChangeDutyCycle(7.5)
        time.sleep(0.5)
        p.ChangeDutyCycle(2.5)
        time.sleep(0.5)

while True:
    if not no_rain.is_active:
        wiper(3)
```

CHAPTER 6: TEST CASES

Intents	Utterance	Action
permission to turn ON vehicle	Main power supply is turned on	Manager press on the button on dashboard by which the relay sensor closes the circuits which turns on the main power supply of the vehicle.
not raining	No action	Rain sensor doesn't sense the rain so no any action taken place.
it starts raining	Automatic wiper turns on	Rain sensor senses the rain and the servo motor start rotating by which automatic wiper will turn on.
it stops raining	Automatic wiper turns on	Rain sensor stop sensing the rain and the servo motor stops rotating by which automatic wiper will turn off.
no obstacle faced while parking	No need to alert driver	Ultrasonic sensor detects the distance of obstacle. If the distance is not minimum, buzzer/alarm won't turn on.
obstacle about to hit car while parking	Alarm/Buzzer alert the driver	Ultrasonic sensor detects the distance of obstacle. If the distance is minimum, microcontroller turns on the buzzer/alarm.
vehicle speed is normal	No any problem detected	Accelerometer find the speed of the vehicle. If speed doesn't cross the limit, no any problem detected. Driver and administrator can view the speed of vehicle at any time.
vehicle overspeed	Warns the driver and notify the administrator	Accelerometer find the speed of the vehicle. If speed crosses the limit, it alerts the driver to slow down and notify the administrator. Driver and administrator can view the speed of vehicle at any time.

Vehicle is inside the geofenced area	No any problem detected	GPS sensor find the live location. If the location doesn't go outside the geofenced area, no need to notify the administrator.
Vehicle move outside the geofenced area	Administrator is notified about this situation	GPS sensor find the live location. If the location goes outside the geofenced area, administrator will be notified about current situation of driver and action can be taken against them.
Driver is drowsy	No need to alert driver	Camera module takes the live footage of the driver and if driver doesn't close eye for more than 5 seconds it means driver is not drowsy. Driver won't be alerted.
Driver is drowsy	Alarm/Buzzer alert the driver and administrator is notified	Camera module takes the live footage of the driver and if driver closes eyes for more than 5 seconds it means driver is drowsy. Driver will be alerted and also administrator will be notified.
permission to turn OFF vehicle	Main power supply is turned off	Manager press off the button on dashboard by which the relay sensor opens the circuits which turns off the main power supply of the vehicle.

CHAPTER 7: RESULTS

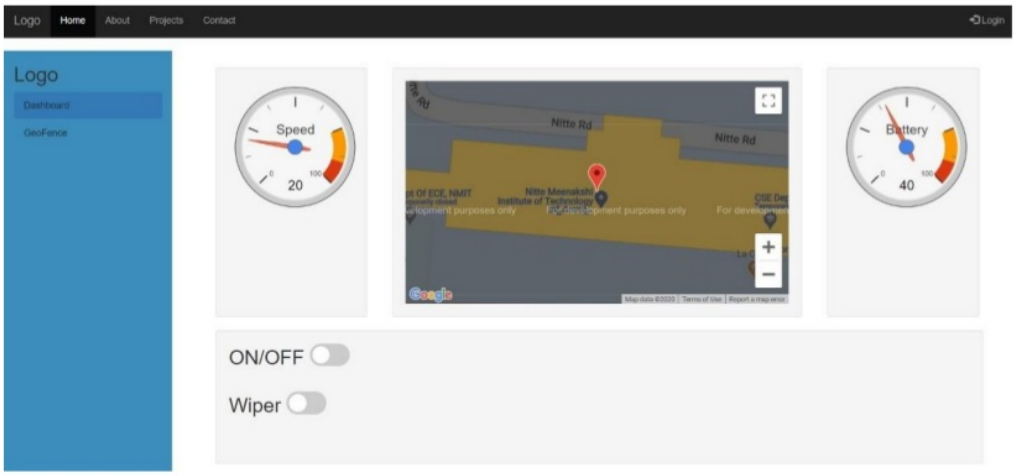

Fig 14: Screenshot showing main dashboard

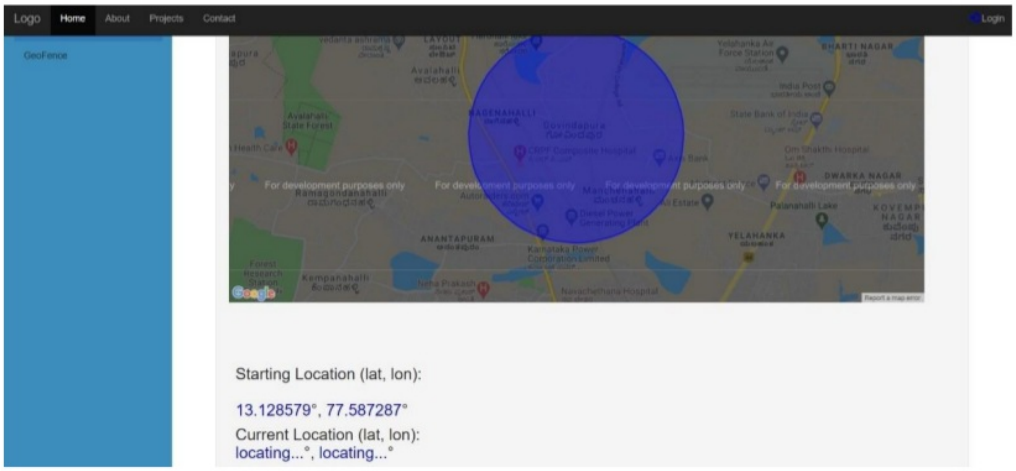

Fig 15: Screenshot showing Geofencing monitoring

CHAPTER 8: IMPACT OF THIS PROJECT TOWARDS SOCIETY/ ENVIRONMENT

Car Fleet Management have its own good and bad impact towards the society or environment. Some of impacts are mentioned below:

- Having a personal vehicle has its advantages because people can save their time by travelling on their own vehicle instead of using public transport. Car Feet Management provides the car for personal use which can satisfy the people's need and can provide ease in the community. Leasing would help in cases where the user wants the vehicle for an extended period but does not want to invest money in buying and maintaining the vehicle.
- Instead of using high number of personal vehicles, if everyone uses the vehicle of fleet it can decrease the pollution in the environment. Also, this project is built on electric vehicle so itself it decreases the pollution in the environment.
- Internet of Things have its own advantages and disadvantages in environment. Since, IoT uses different kind of RF waves it affects the environment. We have seen the disappearing of birds nowadays as compared to past due to increase of mobile towers. So, IoT also uses these kinds of waves which can have an impact on the environment.

CHAPTER 9: CONCLUSION

This project aims to target the companies like Zoom car where they will be able to track and monitor their vehicle in real time and help to reduce the cost for maintenance since the vehicle information can be visible using IoT in the dashboard.

It will reduce the work of the fleet manager to a great extent as all activities are digitalized and automated. Every sensor acquires different type of data and sends it to the monitoring system. In monitoring system Firebase stores the data and the microcontroller processes all the data and again stores it in the firebase. After processing all the data, the information is visualized in the dashboard. Dashboard displays GPS location of the vehicle, Geofencing, Driver Drowsiness, Vehicle ON/OFF status, Wiper status. It just needs some manager to stay at backend and monitor the status of the vehicle. With the successful implementation of this system it can prove highly advantageous to organization. Regular update on the system after implementation can upgrade this proposed system and can be one of the best Car Feet Management System in future.

CHAPTER 10: REFERENCES

[1] Thiyagarajan Manihatty Bojan, Umamaheswaran Raman Kumar, and Viswanathan Manihatty Bojan, *"An Internet of Things based Intelligent Transportation System"*, 2014 IEEE International Conference on Vehicular Electronics and Safety (ICVES), December 16-17, 2014. Hyderabad, India [Available: IEEE Explore Org]

[2] Mahaveer Penna, Shivashankar, B Arjun, K R Goutham, Lohith N Madhaw, Kumar G Sanjay, *"Smart fleet monitoring system using Internet of Things (IoT)"*, 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 19-20 May 2017, Bangalore, India [Available: IEEE Explore Org]

[3] Channakeshava Gowda V R, Gopalakrishna K, *"Real Time Vehicle Fleet Management and Security System"*, 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 10-12 December 2015, Trivandrum [Available: IEEE Explore Org]

[4] Challa Yashwanth and Jyoti Singh Kirar, *"Driver's Drowsiness Detection"*, 2019 IEEE Region 10 Conference (TENCON), Year 2019, Hyatt Kochi Bolgatti, Kerela, India [Available: IEEE Explore Org]

[5] R. Srinivasan, A. Sharmili, Dr. S. Saravanan, D. Jayaprakash, *"Smart Vehicles with Everything"*, 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Year 2016, Salem, India [Available: IEEE Explore Org]

[6] Sarah Wilkinson, "Everything you need to know about Fleet Management" *chevinfleet.com*, para. 1, Sep. 25, 2018. [Online]. Available:https://www.chevinfleet.com/au/news/what-is-fleet-management-software. [Accessed Feb. 12, 2020].

[7] "HC-SR04 Ultrasonic Sensor" *components101.com,* para. 1 to 5, Sep. 18, 2017. [Online]. Available:https://components101.com/ultrasonic-sensor-working-pinout-datasheet. [Accessed Mar. 20, 2020].

CHAPTER 11: SELF ASSESSMENT OF PO-PSO ATTAINMENT

APPENDIX 1 Paper/Patents with Certificates

APPENDIX 2 Plagiarism Report

Prashant Shah

ORIGINALITY REPORT

11%	7%	6%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	V R Channakeshava Gowda, K Gopalakrishna. "Real time vehicle fleet management and security system", 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2015 Publication	2%
2	stackoverflow.com Internet Source	2%
3	www.manihattysemiconductors.com Internet Source	1%
4	Challa Yashwanth, Jyoti Singh Kirar. "Driver's Drowsiness Detection", TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), 2019 Publication	1%
5	Submitted to SASTRA University Student Paper	1%
6	www.semakudu.com Internet Source	1%
7	Mahaveer Penna, Shivashankar, B Arjun, K R	

	Goutham, Lohith N Madhaw, Kumar G Sanjay. "Smart fleet monitoring system using Internet of Things(IoT)", 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017 Publication	1%
8	icact.org Internet Source	1%
9	iotdesignpro.com Internet Source	<1%
10	R. Srinivasan, A. Sharmili, S. Saravanan, D. Jayaprakash. "Smart vehicles with everything", 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016 Publication	<1%
11	Submitted to University of Balamand Student Paper	<1%
12	Submitted to PSB Academy (ACP eSolutions) Student Paper	<1%
13	Submitted to Jawaharlal Nehru Technological University Kakinada Student Paper	<1%
14	Submitted to University of Winchester Student Paper	<1%

15	community.blynk.cc Internet Source	<1%
16	Submitted to SRM University Student Paper	<1%
17	how2electronics.com Internet Source	<1%
18	wikiserver.infomerce.es Internet Source	<1%
19	Submitted to Bolton Institute of Higher Education Student Paper	<1%
20	Submitted to Informatics Education Limited Student Paper	<1%
21	Submitted to Higher Education Commission Pakistan Student Paper	<1%
22	Submitted to Fullerton College Student Paper	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		