**Lourdes Matha College of Science and Technology**

**(Managed By Archdiocese Of Changanacherry)**

**(Affiliated To APJ Abdul Kalam Technological University, Kerala)**

**Kuttichal, Thiruvananthapuram-695574**

**Department of Computer Applications**

****

**Certificate**

**This is to certify that the project work entitled “BE SAFE BREATH ANALYSER” is a bonafide record of the work done by Mr JAYAKRISHNAN J U, Reg No LMC17MCA003, student of Department Computer of Applications, Lourdes Matha College of Science & Technology, Kuttichal, Thiruvananthapuram, affiliated to APJ Abdul Kalam Technological University, Kerala during the academic year 2019-2020 from January 2019 to May 2020 in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications from APJ Abdul Kalam Technological University, Kerala.**

**Internal Guide** **Date:** **Head of the Department**

**Internal Examiner** **External Examiner**

**ACKNOWLEDGEMENT**

**“**It is not possible to prepare a project report without the assistance & encouragement of other people. This one is certainly no exception**”**

On the very outset of this report. I would like to extend my sincere & heartfelt obligation towards all the personages who have helped me in this endeavour without their active guidance, help, operation & encouragement, I would not have made headway in the project.

It is a great pleasure to express my sincere gratitude to **Rev.Dr.Tomy Joseph Padinjareveetil** ,Director and **Prof.Dr.Mohanlal PP** ,Principal , Lourdes Matha College Of Science and Technology for permitting to do this project with the fullest spirit .

I am ineffably indebted to **Prof. Justin G Russel** for conscientious guidance and encouragement to accomplish this assignment.

I would like to express the deepest appreciation to **Prof. Sherin Joseph** for her continuous help and support for development of this project at all phases.

I am extremely thankful and pay my gratitude to **Prof.Selma Joseph** Head of the department of Computer Applications,Lourdes Matha College Of Science and Technology, for valuable guidance and support on completion of this project in its presently.

I extend my gratitude to Lourdes Matha College Of Science And Technology for giving me this opportunity.

I also acknowledge with a deep sense of reverence, my gratitude towards my parents and member of my family, who has always supported me morally as well as economically**.**

At last but not least gratitude goes to all of my **friends** who directly or indirectly helped me to complete this project report.

Any omission in this brief acknowledgement does not mean lack of gratitude.

**DECLARATION**

I undersigned hereby declare that the project report “BE SAFE – BREATH ANALYSER”, submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Application of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Prof. Sherin Joseph**. This submission represents my ideas in my own words and I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute or the University.

**ABSTRACT**

Most of these days, we hear lot of accidents due to drunken driving. Drunken drivers will not be in stable condition and so rash driving is an inconvenience for other road users and also a question of life and death for the drunken driver and for others.

There are many different types of accidents which occur in daily life. Accidents may occur due to many reasons like brake fail, reckless driving, vehicle defects etc. Most often road accidents are caused by drivers who had consumed alcohol. Though there are laws to punish drunken drivers they cannot be fully implemented. Because traffic police cannot stand on every road to check each and every vehicle driver whether he/she has drunk or not. This can be a major reason for accidents. So, there is a need for an effective system to check drunken drivers.

The aim of our project is to make human driving safer and to overcome these accidents, for that we have designed an embedded system for implementing an efficient alcohol detection system that will be useful to avoid accidents. Therefore, we implemented a system that will initially check whether the person has drunken or not, by using the MQ3 (Alcohol) sensor.

The breath analyser senses the amount of alcohol present in the breath of a person and reports if it is beyond the legal limit. The device can be connected to any smartphone via Bluetooth and communication with the online API can be done using the internet connection of the smartphone. This will ensure the holistic safety of the rider at all times. This device helps in instant capturing of image of the driver and also creation of a chart of his alcohol consumption. All data is sent to the cloud and a pdf file is generated to show the penalty he has to pay and if the next time he gets caught the previous history is also obtained and thus the penalty can be doubled. It also helps in getting the location as well as other details about the driver and vehicle etc. We can also identify the amount of alcohol present in the persons body as well as the type of alcohol which was consumed.

**CHAPTER 1**

**INTRODUCTION**

**1.1 GENERAL BACKGROUND**

Motorcycles and bikes form an integral part of personalized transportation in India. However, unfortunately, it also involves innumerable accidents and subsequent loss of lives. Every year, about 300,000 teenagers go to the emergency department because of bike injuries, and at least 10,000 teenagers have injuries that require a few days in the hospital. Statistics say, motorcycle deaths accounted for 15 % of all motor vehicle crash deaths in 2015 and were more than double the number of motorcyclist deaths in 1997. Through an ONEISS survey conducted by the Department of Health, it was found that 90% of the motorcycles rider killed in accidents were not wearing a helmet at the time of impact. This, along with drunken driving are a major reason of accidents.

Drunk driving is a big problem in every part of the nation. In 2015 alone, over 10,000 traffic fatalities were linked directly to drivers who had blood alcohol levels above the legal limit. Many accidents happen due to the carelessness on the part of driver. Many drivers drink and drive which is a criminal offence. Such drivers are a menace to society and should be apprehended quickly. Though the country has laws to check drunken driving but its effective implementation is still to be worked upon and in some cases even questionable. For such purpose we are designing a system which will assist the traffic police officers to determine whether he/she is fit to drive or not. This system is basically an Embedded System which is combination of both software and hardware which can perform some specific functions using Micro Controller Arduino Nano and MQ3 Gas Sensor along with a smart phone.

The Alcohol sensor on detecting the alcohol concentration will give the analog resistive output to the microcontroller then further alcohol detection message will be displayed on the app. An onboard alcohol sensor also analyzes the breath of the rider to detect if the current intoxication level is above the legal threshold. Which helps in the identification of the driver’s alcohol consumption ratio and penalize him/her easily.

**1.2 OBJECTIVE AND SCOPE**

In this project we are mainly focusing on the safety of vehicle drivers. This project is used as an "Alcohol detector". Alert is sent on an Application installed on Android mobile. This project can be used in Colleges, University campus, Industries and companies. The main objective of the project is to detect whether the person has consumed the alcohol or not. Alcohol sensor is used to detect the alcohol. Microcontroller sends alert to Android mobile using Bluetooth transmitter. Person has to breathe out in front of Alcohol sensor.

Generally, this project should be installed on the entrance gate of college or company. If the alcohol percentage is more than the threshold value then microcontroller turns on the Buzzer immediately. The breath analyser senses the amount of alcohol present in the breath of a person and reports if it is beyond the legal limit. The device can be connected to any smartphone via Bluetooth and communication with the online API can be done using the internet connection of the smartphone. This will ensure the holistic safety of the rider at all times. This device helps in instant capturing of image of the driver and also creation of a chart of his alcohol consumption. All data is sent to the cloud and a pdf file is generated to show the penalty he has to pay and if the next time he gets caught the previous history is also obtained and thus the penalty can be doubled. It also helps in getting the location as well as other details about the driver and vehicle etc. We can also identify the amount of alcohol present in the person’s body as well as the type of alcohol which was consumed.

Alcohol sensor MQ3 is suitable for detecting alcohol concentration just like your common Breath analyzer. It has a high sensitivity and fast response time, Sensor provides an analog resistive output based on alcohol concentration which is given to inbuilt ADC of microcontroller. Along with this we use smart phone with consists of an android application that focuses on capturing image of the drunken driver as well as collection of his licence details and vehicle data, also the location details in which the person was caught red-handed. We can also obtain the amount of alcohol present in the persons blood instantly as well as a graph of his alcohol consumption is generated with consist of the level of alcohol ratio in his blood. A pdf file is automatically generated which consist of all these details and a copy of this data is stored in the cloud for future verification.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 STUDY OF SIMILAR WORK**

In the literature, I found various alcohol detection systems but with different approach and proposed solution.

Dai [1] proposed” Mobile phone based drunk driving detection”. This system requires a mobile phone. It should be installed in the vehicle with orientation. A program should be installed in that phone it compares the sensor readings with the real time driving test cases. If there is any presence of drunk driving then automatically the phone will alert the driver or sends the alert message to the police/family.

Bhuta [2] proposed “Alcohol Detection and Vehicle Controlling”. Arduino is used in this system. An alcohol detector and GPS module, GSM module are connected to the Arduino board. When the alcohol level in the driver’s body crosses a particular limit vehicle’s ignition will off and GPS module finds out the location and alert will be sent to the police and family using GSM module.

Gupta [3] proposed “Alcohol Detection with Vehicle Controlling”. This system is mainly used for preventing accidents that are occurring due to drunken driving. An alcohol detector is connected to the PCB (Printed Circuit Board).When alcohol level crosses a permissible limit ignition of the vehicle is stopped. This system should be installed inside the vehicle.

Goswami [4] proposed “Alcohol Detection with Vehicle Controlling”. This system needs an android phone with some sensors and program installed on android phone. This program matches the current sensor readings with the values that are already present in the system. If any match found then an alert message will be sent along with the location.

Phalak [5] proposed “Smartphone and Sensor Based Drunk Driving Prevention System”. According to this system DUI (Driving under Influence) of alcohol is the major cause of accidents. So they proposed a system to find the driver’s condition based on the real driving test cases. A specially designed hardware consisting of sensors and a mobile are installed in the vehicle. A program is installed in hardware and mobile phone. When a person is driving the vehicle sensor readings are compared to the real time test case values. If there is any presence of DUI of alcohol alert message will be sent.

Sarkar [6] proposed “A Real Time Embedded System Application for Driver Drowsiness and Alcoholic Intoxication Detection”. This system uses embedded system. This system consists of 5 megapixel camera with embedded system board raspberry. Raspberry pi is interfaced with another arduino board which is used for some tasks like alarm notifications and ignition lock .It performs these tasks if and only if it receives a message from raspberry pi about the presence of alcohol.

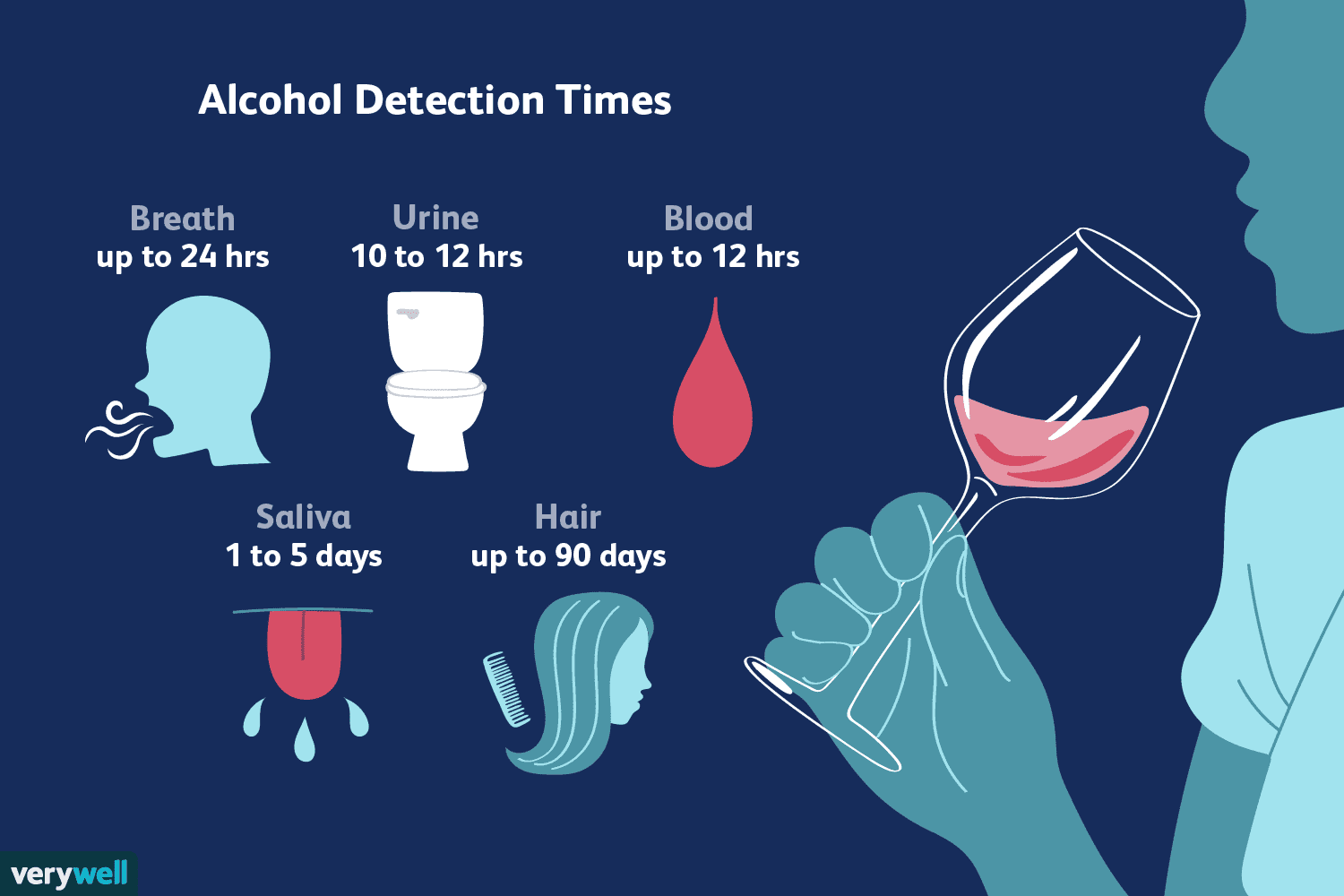
Savania [7] proposed “Alcohol Detection and Accident Prevention of Vehicle”. In this an alcohol sensor is placed in the vehicle to detect the alcohol. If the alcohol gases are detected then for every 5 minutes a message is sent to their relatives. In this an arm7 microcontroller is used and it is connected to GSM and GPS. The GPS is used to track the location of the vehicle and message will be sent using GSM module.

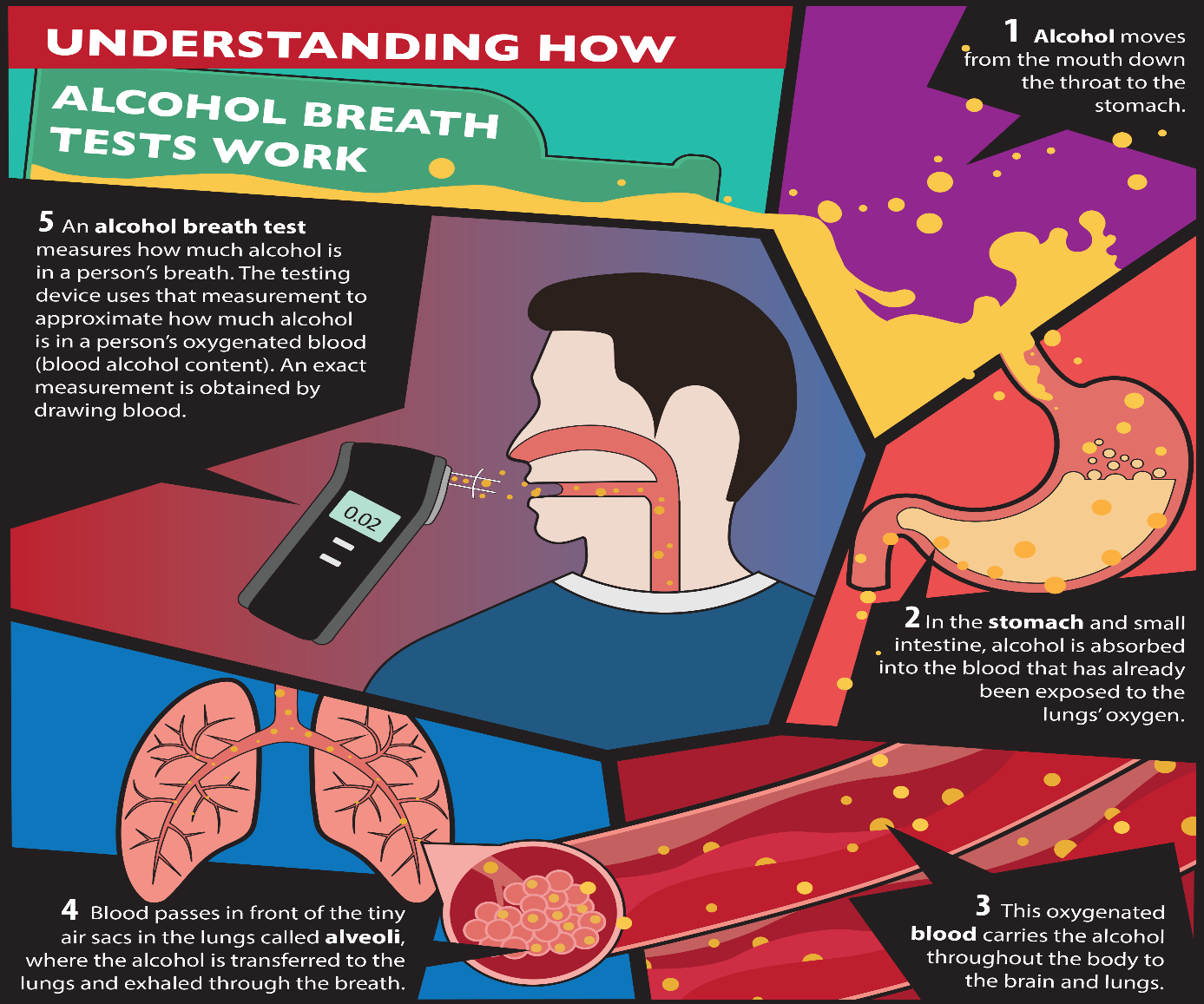
Deshmukh [8] proposed “Driver fatigue Detection Using Sensor Network”. This system consists of sensors that are directly faced towards driver’s face. This system monitors the driver eyes to check whether the driver is sleeping or not by eye blink sensor and detect their pulse from fingers by using LED and LDR. Analyze the sensor readings and find the fatigue level.

Wilhelm Von Rosenberg et al [9] has Proposed a smart helmet with embedded sensors for cycling and moto racing to monitor both vital signs and the electroencephalogram (EEG) simultaneously. They have embedded multiple electrode within a standard helmet and a respiration belt around the thorax for validation and a reference ECG from the chest. Also a multivariate R-peak detection algorithm has been applied to get data from real life noisy environment.

Sreenithy Chandran et al [10] monitored the value received from accelerometer embedded in helmet and detects an accident by analysing those values and sends an emergency notification to contacts with Global positioning system location.

C. J. Behr et al [11] has developed a smart helmet for the mines industry keeping focus air quality, helmet removal and collision. It detects the presence of hazardous gases like CO, SO2, NO2 and detects the helmet removal using an off-the shelf IR sensor.





**2.1.1 Existing System**

Breath analyzers do not directly measure blood alcohol content or concentration, which requires the analysis of a blood sample. They estimate BAC indirectly by measuring the amount of [alcohol](https://en.wikipedia.org/wiki/Alcohol_(drug)) in one's breath. Small hand-held breath analyzers are not reliable enough to provide evidence in court but reliable enough to justify an arrest. Larger breath analyzer devices found in [police stations](https://en.wikipedia.org/wiki/Police_station) can then be used to produce court evidence. The test device provides numerical blood alcohol content (BAC) readings, although in some cases, the device has "pass/fail" indication. If the same person caught for multiple times but it cannot be determined using existing system. Also the existing system cannot quickly identify the vehicle as well as driver details as the need to contact further higher authorities more getting these details.

In existing system the driver of the vehicle is taken to the hospital for futher examination and collection of alcohol blood content data .This is a time consuming process as well as if the test is delayed the ratio of alcohol content in blood also decreases and it will become a loophole for the culprit for escape from the case. Also the alcohol type consumed is also unidentified.

**2.1.2 Drawbacks of Existing System**

* Lack of safe and security
* Less Efficient
* Implementation Issues
* High Expenses
* Difficulty in Maintenance
* More human efforts
* It is very tedious and time consuming

**CHAPTER 3**

**OVERALL DESCRIPTION**

**3.1 PROPOSED SYSTEM**

In the proposed system we propose mechanisms that can detect whether the person has over-consumed alcohol. Alcohol sensor is used to identify the level of alcohol in the person. A person has to breathe out in front of an Alcohol sensor. The microcontroller sends alert to Android mobile using Bluetooth transmitter. Generally breathing analyzer project should be installed on the entrance gate of college or company. If the alcohol percentage is more than the threshold value then microcontroller turns on the Buzzer immediately. Then an alert message is sent to Bluetooth encoder and then it sends it to Bluetooth transmitter.

The breath analyser senses the amount of alcohol present in the breath of a person and reports if it is beyond the legal limit. The device can be connected to any smartphone via Bluetooth and communication with the online API can be done using the internet connection of the smartphone. This will ensure the holistic safety of the rider at all times. This device helps in instant capturing of image of the driver and also creation of a chart of his alcohol consumption. All data is sent to the cloud and a pdf file is generated to show the penalty he has to pay and if the next time he gets caught the previous history is also obtained and thus the penalty can be doubled. It also helps in getting the location as well as other details about the driver and vehicle etc. We can also identify the amount of alcohol present in the persons body as well as the type of alcohol which was consumed. Breath analyzers are actively considered as the best solution so far to check a person is drunk or not, given that they are used by police force of every country.

**3.2 Sensor Based Breath Analyzing**

Whenever a person drinks alcohol, its shows up in the breath as it gets absorbed from the mouth, throat, stomach and intestines directly into the bloodstream. Digestion of Alcohol does not occur upon absorption, nor does it chemically change in the bloodstream. As the blood moves through the lungs, some of the alcohol moves across the membranes of the lung's air sacs (alveoli) into the air, because alcohol will evaporate from a solution that is, it is volatile. The concentration of the alcohol in the alveolar air is related to the concentration of the alcohol in the blood. As the alcohol in the alveolar air is exhaled, it can be detected by the breath alcohol testing device. Thus when a person breaths into the breath analyzer, it is easy to check if the person is drunk. The working of breath analyzer depends on detecting of level ethanol in breath, and as per the chemical composition of alcohol, the main component of alcohol is ethanol. In general, the breath of a person consists of no level of ethanol.

**3.2 FEATURES OF PROPOSED SYSTEM**

* Wireless Communication
* Reduction of Manpower
* Getting Alcohol Consumption Details
* History of Previous Penalty Record.
* Cost Effective
* Efficient
* Can Implement Easily
* Safe and Secure
* Easy to maintain
* Can easily add advanced technologies
* Light Weight
* Ease of Use

**3.3 FUNCTIONS OF PROPOSED SYSTEM**

* **Portability**: Can be used and carried anywhere at anytime.
* **Reliable:** The results produced by examination are accurate.
* **Light Weight**: Using Just sensor and microcontroller making system much smaller and efficient.
* **Low Cost:** With Just Minimum number of hardware makes this system is cheaper and more affordable.
* **Adaptive:** This proposed system can be easily adapted to any new technology.
* **Data Collection:** We can obtain the driver details along with his/her image.
* We design and implement a drunk driving detection system with specialized hardware and smart mobile phone. The system is reliable, non-intrusive, lightweight, and power-efficient and easily integrate able in any system.

**3.4 REQUIREMENT SPECIFICATION**

System analyst talk to a variety of persons to gather details about the business process and their opinions of why things happen as they do and their ideas for changing the process. These can be done through questionnaire, detailed investigation, observation, collection of samples etc. As the details are collected, the analyst studies the requirements data to identify the features of the new system must have, including both the information the system should produce and operational features such as processing controls, response times and input-output methods.

Requirements specification simply means, “Figuring out what is to be made before making it”. It determines what people need before starting to develop a product for them. Requirement definition is the is the activity of translating the information gathered in to a document that defines a set of requirements. These should reflect what consumer wants.

The main requirements of the BE SAFE (Breath Analyser) system are Arduino Nano,MQ3 Alcohol Sensor , Bluetooth Module , 5V Power Supply and a Smart Phone.

**3.5 FEASIBILITY ANALYSIS**

The initial investigation points to be question whether the project is feasible. The feasibility study concerns with the considerations made to verify whether the system fit to be developed in all terms. Once the idea to develop software is put forward, the question that rises first will pertain to be the feasibility aspects. Feasibility study is a test of proposed system regarding its efficiency, impact on the organization, ability to meet the need of users and effective use of resources.

Thus, when a new project is proposed, it normally goes through a feasibility study before it is approved for development. A feasibility study is conducted to select the best system that meets the system performance requirements. This entitles an identification description, an evaluation of candidate system and the selection of the best system for the job.

During system analysis, a feasibility study of the proposed system was carried out to see whether it was beneficial to the organization. Three key considerations that are involved in the feasibility study. They are,

Technical Feasibility

Economic

Behaviour Feasibility

Operational Feasibility

**3.5.1 Technical Feasibility**

Technical Feasibility centres on the existing computer system hardware, software, etc. and to some extent how it can support the proposed addition. This involves financial considerations to accommodate technical enhancements. Technical support is also a reason for the success of the project. The techniques needed for the system should be available and it must be reasonable to use. Technical Feasibility is mainly concerned with the study of function, performance, and constraints that may affect the ability to achieve the system. By conducting an efficient technical feasibility, we need to ensure that the project works to solve the existing problem area.

Since the project is designed using Android as programming language. It is very efficient and user friendly. Here we are using Arduino Nano micro controller to feed the program and the readings from the mq3 sensor are send to the mobile phones using Bluetooth which easy to use and maintain.

**3.5.2 Economic Feasibility**

The role of interface design is to reconcile the differences that prevail among the software engineer’s design model, the designed system meets the end user requirement with economical way at minimal cost within the affordable price by encouraging more of proposed system. Economic feasibility is concerned with comparing the development cost with the income/benefit derived from the developed system. In this we need to derive how this project will help the management to take effective decisions.

Economic Feasibility is mainly concerned with the cost incurred in the implementation of the project. Since this project is developed using Android which is more commonly available and it is an open source application Android Studio IDE helps in easy implementation of the system.

This project has only a single sensor which is available at low cost in the market.Now a days smart phone is cheap and used by everyone. Also, the price of micro controller Arduino Nano is affordable. The installation cost of Bluetooth, Arduino IDE are also free.

The system once developed must be used efficiently. Otherwise there is no meaning for developing the system. For this a careful study of the existing system and its drawbacks are needed. The user should be able to distinguish the existing one and Proposed one, so that one must be able to appreciate the characteristics of the proposed System, the manual one is not highly reliable and also is considerably fast. The proposed system is efficient, reliable and also quickly responding.

**3.5.3 Behaviour Feasibility**

Proposed projects are beneficial only if they can be changed in to information system that will meet operation requirement of the organization. People are inherently resistant to change and computers have been known to facilitate changes. An estimate should be made of how strong reaction the user staff is likely to have towards the development of a computerized system.

The drunk driving behavior is studied under single solution: the analyzing of breath. We will be experimenting the hardware with different types of breaths in real time to understand the difference between the normal condition and the drunk driving condition. The data will be collected using different types of alcohols so that an accurate range of values are generated which can then be applied to the system for appropriately evaluating the current condition and to avoid any false positives or negatives.We obtain the driver data along with alcohol consumption details which helps in penalizing him/her.

**3.5.3 Operational Feasibility**

There is not much difficulty in implementing the system. The proposed system is effective, user friendly and functionally efficient. The user of the system must be unaware of the internal working of the system so that the user will not face any problems running the system. In our system we are using Arduino Nano microcontroller and MQ3 sensor. We can extend or add any features to the system easily.

The sensor will give accurate value according to the programs that are fed in to the micro controller. The user can easily use the system and app. There is no need to worry about the internal procedures of the system.

**CHAPTER 4**

**OPERATING ENVIOURNMENT**

**4.1** **HARDWARE REQUIREMENTS**

Processor : Intel Core i5 or AMD Ryzen 5

RAM : 8 GB

Hard Disk : 500GB

External Hardware : Arduino Nano , Alcohol Sensor-MQ3 , Bluetooth Module , Smart Phone

**4.2** **SOFTWARE REQUIREMENTS**

Operating System : Windows 10

Language : Android

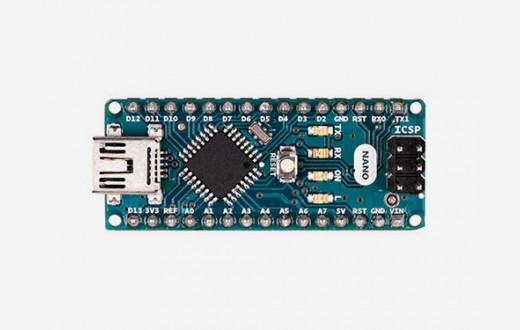
Server :

IDE : Android Studio , Arduino IDE

**4.3 TOOLS AND PLATFORMS**

**4.1.1 Arduino Nano**

ARDUINO NANO



The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.

### **Power**

 The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

### **Memory**

 The ATmega328 has 32 KB, (also with 2 KB used for the bootloader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

### **Input and Output**

 Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

 AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

### **Communication**

 The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. To use the SPI communication, please see ATmega328 datasheet.

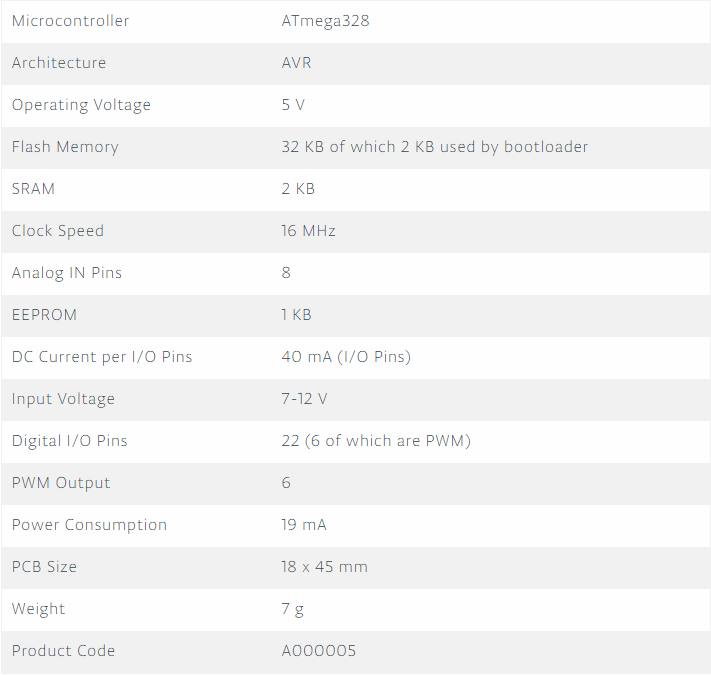
### **Programming**

The Arduino Nano can be programmed with the Arduino software ([download](https://www.arduino.cc/en/main/software)). Select "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). The ATmega328 on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.

### **Automatic (Software) Reset**

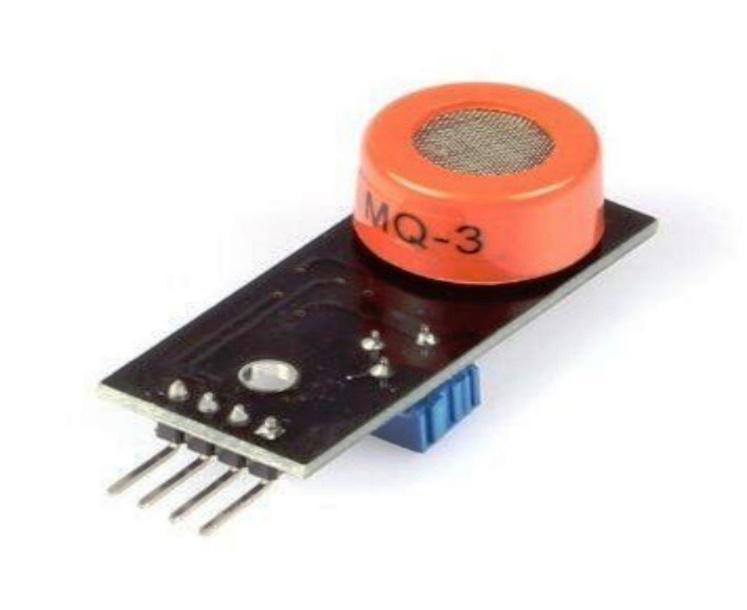
Rather then requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data

**TECHNICAL SPECIFICATIONS**



**4.1.2 Alcohol Sensor Module – MQ3**

# Alcohol Sensor Module - MQ3



5V operation, Analog+Digital Output, threshold preset adjustable, Useful in making Breath Analyzer .

The alcohol sensor we will use is the MQ-3 sensor. This is a sensor that is not only sensitive to alcohol, particularly ethanol, which is the type of alcohol which is found in wine, beer, and liquor.

This type of sensor circuit can be used as a breathalyzer to check a person's blood alcohol level. Just as we exhale carbon dioxide when we breathe out, we also will breathe out some alcohol if we have alcohol in our blood. Any alcometer device can measure this alcohol content.

The more ethanol in your blood, the more there is in the air on exhalation. This alcohol content gives a good indication for if a person is drunk and how drunk they are.

The amount of alcohol exhaled into the air is proportional to the amount of alcohol which will be found in a person's blood. Alcometers use a built-in formula to estimate blood alcohol content from exhaled air alcohol content.

For different countries, the level of alcohol in the blood that defines a person as over the limit for driving varies. The range ranges from 0.01 to 0.10. Most countries have a limit of about 0.05. For example, Greece, Greenalnd, and Iceland all have limits of 0.05. Canada has a higher limit set at 0.08. In the United States, it is also 0.08. This means that if the alcometer reading measures above this, the person can receive a DUI.

For our circuit, it can function as an alcometer so that we get an estimate of a person's blood alcohol level.

The alcohol sensor we will use is the MQ-3 sensor. This is a sensor that is not only sensitive to alcohol, particularly ethanol, which is the type of alcohol which is found in wine, beer, and liquor.

This type of sensor circuit can be used as a breathalyzer to check a person's blood alcohol level. Just as we exhale carbon dioxide when we breathe out, we also will breathe out some alcohol if we have alcohol in our blood. Any alcometer device can measure this alcohol content.

The more ethanol in your blood, the more there is in the air on exhalation. This alcohol content gives a good indication for if a person is drunk and how drunk they are.

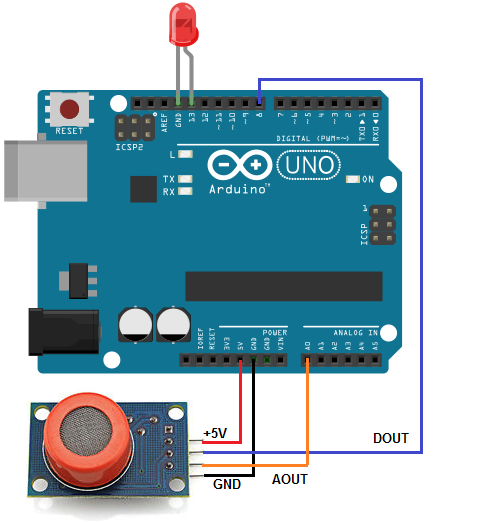
The amount of alcohol exhaled into the air is proportional to the amount of alcohol which will be found in a person's blood. Alcometers use a built-in formula to estimate blood alcohol content from exhaled air alcohol content.

For different countries, the level of alcohol in the blood that defines a person as over the limit for driving varies. The range ranges from 0.01 to 0.10. Most countries have a limit of about 0.05. For example, Greece, Greenland, and Iceland all have limits of 0.05. Canada has a higher limit set at 0.08. In the United States, it is also 0.08. This means that if the alcometer reading measures above this, the person can receive a DUI.

For our circuit, it can function as an alcometer so that we get an estimate of a person's blood alcohol level.

### MQ-3 Alcohol Sensor Circuit Schematic

The alcohol sensor circuit we will build with an MQ-3 sensor integrated with an arduino is shown below.



The connections are pretty basic.

To connect the sensor, there are 4 leads. 2 of them are for power. The +5V terminal of the sensor connects into the 5V terminal of the Arduino board. The GND terminal of the sensor connects into the GND terminal of the Arduino. This establishes power for the sensor.

The other 2 connections are the analog and digital output of the sensor. These connect to analog pin A0 and digital pin D8, respectively.

## Features

* 5V operation
* Simple to use
* LEDs for output and power
* Output sensitivity adjustable
* Analog output 0V to 5V
* Digital output 0V or 5V
* Low Cost
* Fast Response
* Stable and Long Life
* Good Sensitivity to Alcohol Gas
* Both Digital and Analog Outputs
* On-board LED Indicator

## Technical Data

* Concentration : 0.05 mg/L ~ 10 mg/L Alcohol
* Operating Voltage : 5V ±0.1
* Current Consumption : 150mA
* Operation Temperature : -10°C ~ 70°C

## Pin Out

* VCC – Input Power Supply
* GND – Supply Ground
* DO – Digital Output
* AO – Analog Output

## Applications

* Vehicle Alcohol Detector
* Portable Alcohol Detector

**4.1.3 Bluetooth Module**

# HC-05 - Bluetooth Module

# HC-05 Bluetooth Module

It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.

It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.

It is IEEE 802.15.1 standardized protocol, through which one can build wireless Personal Area Network ([PAN](https://en.wikipedia.org/wiki/Personal_area_network)). It uses frequency-hopping spread spectrum ([FHSS](https://en.wikipedia.org/wiki/Frequency-hopping_spread_spectrum)) radio technology to send data over air.

It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).

HC-05 is a Bluetooth module which is designed for wireless comunication. This module can be used in a master or slave configuration.

# **Pin Description**

# Bluetooth Module Pin Description

Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.

It has 6 pins,

1.  **Key/EN**: It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.

HC-05 module has two modes,

1.  **Data mod**e: Exchange of data between devices.

2.  **Command mode**: It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port is used.

3.  **VCC**: Connect 5 V or 3.3 V to this Pin.

4.  **GND**: Ground Pin of module.

5.  **TXD**: Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)

6. **RXD**: Receive data serially (received data will be transmitted wirelessly by Bluetooth module).

7.  **State**: It tells whether module is connected or not.

**HC-05 module Information**

HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds.

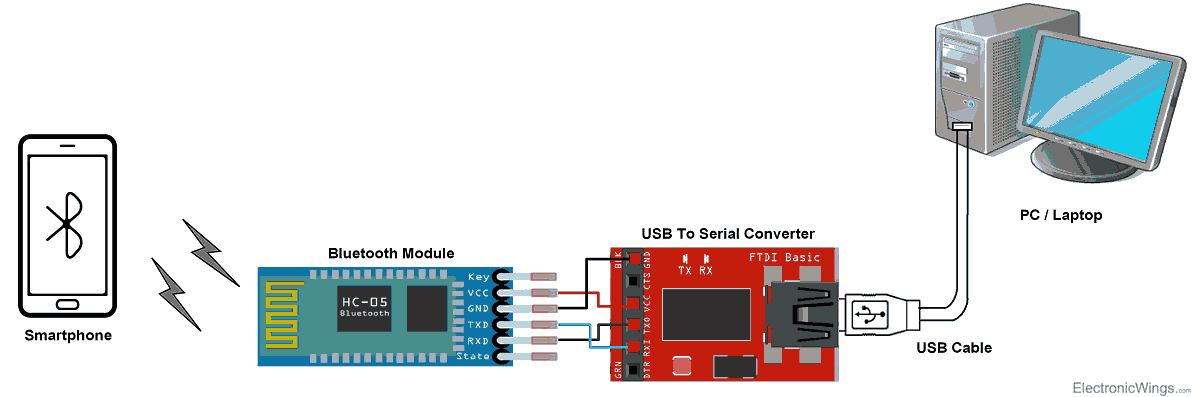
This module works on 3.3 V. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulator.

As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, no need to shift transmit level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.

# **Bluetooth communication between Devices**

 E.g. Send data from Smartphone terminal to HC-05 Bluetooth module and see this data on PC serial terminal and vice versa.

To communicate smartphone with HC-05 Bluetooth module, smartphone requires Bluetooth terminal application for transmitting and receiving data. You can find Bluetooth terminal applications for android and windows in respective app. store.



**Bluetooth Module Serial Interface**

So, when we want to communicate through smartphone with HC-05 Bluetooth module, connect this HC-05 module to the PC via serial to USB converter.

Before establishing communication between two Bluetooth devices, 1st we need to pair HC-05 module to smartphone for communication.

**Pair HC-05 and smartphone**:

Search for new Bluetooth device from your phone. You will find Bluetooth device with “HC-05” name.

Click on connect/pair device option; default pin for HC-05 is 1234 or 0000.

After pairing two Bluetooth devices, open terminal software (e.g. Teraterm, Realterm etc.) in PC, and select the port where we have connected USB to serial module. Also select default baud rate of 9600 bps.

In smart phone, open Bluetooth terminal application and connect to paired device HC-05.

It is simple to communicate, we just have to type in the Bluetooth terminal application of smartphone. Characters will get sent wirelessly to Bluetooth module HC-05. HC-05 will automatically transmit it serially to the PC, which will appear on terminal. Same way we can send data from PC to smartphone.

**Command Mode**

* When we want to change settings of HC-05 Bluetooth module like change password for connection, baud rate, Bluetooth device’s name etc.
* To do this, HC-05 has AT commands.
* To use HC-05 Bluetooth module in AT command mode, connect “Key” pin to High (VCC).
* Default Baud rate of HC-05 in command mode is 38400bps.
* Following are some AT command generally used to change setting of Bluetooth module.
* To send these commands, we have to connect HC-05 Bluetooth module to the PC via serial to USB converter and transmit these command through serial terminal of PC.

**4.1.3 Smart Phone**



A smartphone is a mobile phone with highly advanced features. A typical smartphone has a high-resolution touch screen display, WiFi connectivity, Web browsing capabilities, and the ability to accept sophisticated applications. The majority of these devices run on any of these popular mobile operating systems: Android, Symbian, iOS, BlackBerry OS and Windows Mobile.

A smartphone is expected to have a more powerful CPU, more storage space, more RAM, greater connectivity options and larger screen than a regular cell phone.

High-end smartphones now run on processors with high processing speeds coupled with low power consumptions. That means, they’ll allow you to play 3D games, browse the Web, update your Facebook account, call, and text much longer than you used to.

In addition to the features mentioned earlier, smartphones are also equipped with innovative sensors like accelerometers or even gyroscopes. Accelerometers are responsible for displaying screens in portrait and landscape mode, while gyroscopes make it possible for games to support motion-based navigation.

The earliest touch screen smartphones used resistive touchscreen displays, which required the use of slender pointing objects known as styli (or stylus in singular form). Most of the later models however, like the iPhone and most Android phones, employ capacitive displays, which feature multi-touch finger gestures.

The following are some of the other key features of a smartphone:

* Internet connectivity.
* A mobile browser.
* The ability to sync more than one email account to a device.
* Embedded [memory](https://searchstorage.techtarget.com/definition/memory-card).
* A hardware or software-based [QWERTY keyboard](https://whatis.techtarget.com/definition/QWERTY-keyboard).
* Wireless synchronization with other devices, such as laptop or desktop computers.
* The ability to download applications and run them independently.
* Support for third-party applications.
* The ability to run multiple applications simultaneously.
* [Touchscreen](https://whatis.techtarget.com/definition/touch-screen).
* [Wi-Fi](https://searchmobilecomputing.techtarget.com/definition/Wi-Fi).
* A [digital camera](https://searchmobilecomputing.techtarget.com/definition/digital-camera), typically with video capability.
* Gaming.
* Unified messaging.
* [GPS](https://searchmobilecomputing.techtarget.com/definition/Global-Positioning-System).



**4.2.1 Android**



Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.

On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.

The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

“Application program interface (API) is a set of routines, protocols, and tools for building software applications. An API specifies how software components should interact and APIs are used when programming graphical user interface (GUI) components.”

## **Why Android ?**



## **Features of Android**

Android is a powerful operating system competing with Apple 4GS and supports great features. Few of them are listed below –

|  |  |
| --- | --- |
| **Sr.No.** | **Feature & Description** |
| 1 | **Beautiful UI**  Android OS basic screen provides a beautiful and intuitive user interface. |
| 2 | **Connectivity**  GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX. |
| 3 | **Storage**  SQLite, a lightweight relational database, is used for data storage purposes. |
| 4 | **Media support**  H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP. |
| 5 | **Messaging**  SMS and MMS |
| 6 | **Web browser**  Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3. |
| 7 | **Multi-touch**  Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. |
| 8 | **Multi-tasking**  User can jump from one task to another and same time various application can run simultaneously. |
| 9 | **Resizable widgets**  Widgets are resizable, so users can expand them to show more content or shrink them to save space. |
| 10 | **Multi-Language**  Supports single direction and bi-directional text. |
| 11 | **GCM**  Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution. |
| 12 | **Wi-Fi Direct**  A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection. |
| 13 | **Android Beam**  A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together. |

## **Android Applications**

Android applications are usually developed in the Java language using the Android Software Development Kit.

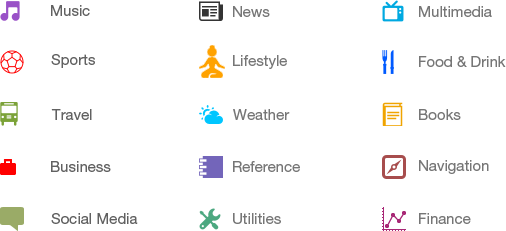
Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play, SlideME, Opera Mobile Store, Mobango, F-droid and the Amazon Appstore.**

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

This tutorial has been written with an aim to teach you how to develop and package Android application. We will start from environment setup for Android application programming and then drill down to look into various aspects of Android applications.

## **Categories of Android applications**

There are many android applications in the market. The top categories are −



**4.2.1 Android Studio**



Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (ADT) as the primary IDE for native Android application development.

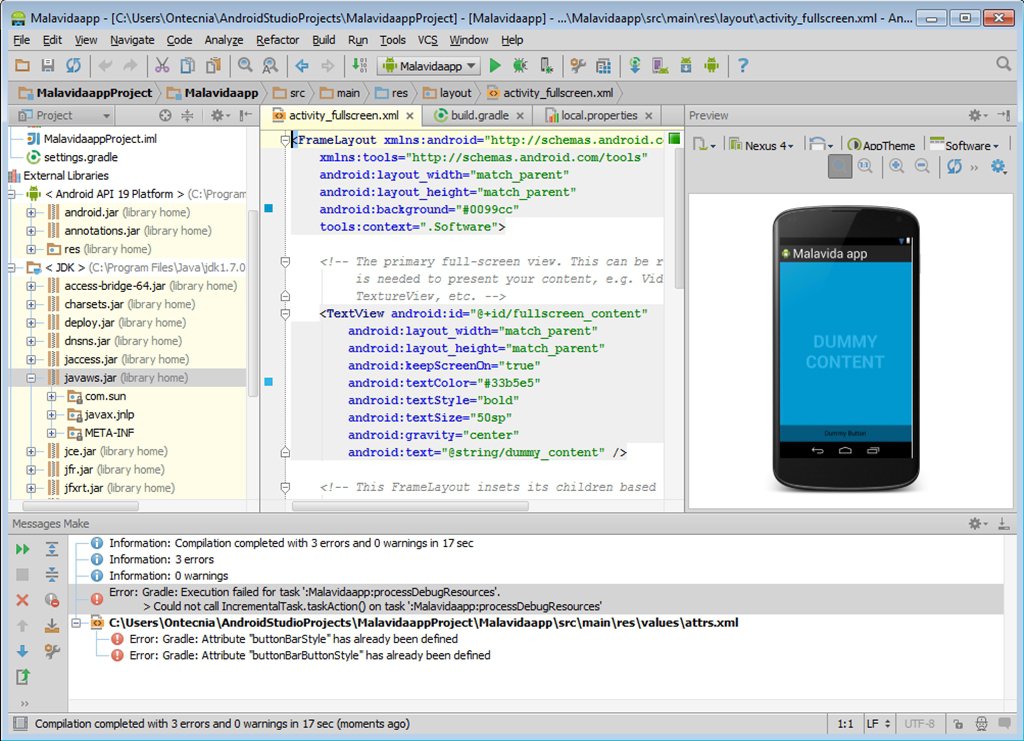
Android Studio was announced on May 16, 2013 at the [Google I/O](https://en.wikipedia.org/wiki/Google_I/O) conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

On May 7, 2019, [Kotlin](https://en.wikipedia.org/wiki/Kotlin_(programming_language)) replaced [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) as Google's preferred language for Android app development. Java is still supported, as is [C++](https://en.wikipedia.org/wiki/C%2B%2B).

Android Studio supports all the same programming languages of [IntelliJ](https://en.wikipedia.org/wiki/IntelliJ) (and [CLion](https://en.wikipedia.org/wiki/CLion" \o "CLion)) e.g. [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), and more with extensions, such as [Go](https://en.wikipedia.org/wiki/Go_(programming_language)); and Android Studio 3.0 or later supports [Kotlin](https://en.wikipedia.org/wiki/Kotlin_(programming_language)) and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects [backport](https://en.wikipedia.org/wiki/Backporting) some Java 9 features. While IntelliJ states that Android Studio is built on supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

## **Features**

* [Gradle](https://en.wikipedia.org/wiki/Gradle)-based build support
* Android-specific [refactoring](https://en.wikipedia.org/wiki/Code_refactoring) and quick fixes
* [Lint](https://en.wikipedia.org/wiki/Lint_(software)) tools to catch performance, usability, version compatibility and other problems
* [Pro-Guard](https://en.wikipedia.org/wiki/ProGuard_(software)) integration and app-signing capabilities
* Template-based wizards to create common Android designs and components
* A rich [layout editor](https://en.wikipedia.org/wiki/Graphical_user_interface_builder) that allows users to drag-and-drop UI components, option to [preview layouts](https://en.wikipedia.org/wiki/WYSIWYG) on multiple screen configurations
* Support for building [Android Wear](https://en.wikipedia.org/wiki/Android_Wear) apps
* Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
* Android Virtual Device (Emulator) to run and debug apps in the Android studio.



**4.2.2 Arduino IDE**

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

* Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
* You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
* Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
* Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
* Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

**CHAPTER 5**

**DESIGN**

**5.1 SYSTEM DESIGN**

System design involves translating system requirements and conceptual design into technical specifications and general flow of processing. After the system requirements have been identified, information has been gathered to verify the problem and after evaluating the existing system, a new system is proposed.

System design is the process of planning of new system or to replace or complement an existing system. It must be thoroughly understood about the old system and determine how computers can be used to make its operations more effective. There are two levels of system design:

* Logical design.
* Physical design.

In the logical design, the designer produces a specification of the major features of the system which meets the objectives. The delivered product of logical design includes current requirements of the following system components:

* Input design.
* Program design
* Output design
* Database design

Physical design takes this logical design blue print and produces the program software, files and a working system. Design specifications instruct programmers about what the system should do. The programmers in turn write the programs that accept input from users, process data, produce reports and store data in files.

**5.1.1 DATA FLOW DIAGRAM/ UML**

i. Block Diagram

A block diagram is a [diagram](https://en.wikipedia.org/wiki/Diagram) of a [system](https://en.wikipedia.org/wiki/System) in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in [hardware design*,*](https://en.wikipedia.org/wiki/Hardware_architecture) [electronic design*,*](https://en.wikipedia.org/wiki/Electronic_design) [software design,](https://en.wikipedia.org/wiki/Software_design) and [process flow diagrams.](https://en.wikipedia.org/wiki/Process_flow_diagram) Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the [schematic diagrams](https://en.wikipedia.org/wiki/Schematic_diagram) and [layout diagrams](https://en.wikipedia.org/wiki/Integrated_circuit_layout) used in electrical engineering, which show the implementation details of electrical components and physical construction.

ii. Flowchart

A flowchart is a type of [diagram](https://en.wikipedia.org/wiki/Diagram) that represents a [workflow](https://en.wikipedia.org/wiki/Workflow) or [process.](https://en.wikipedia.org/wiki/Process) A flowchart can also be defined as a diagrammatic representation of an [algorithm,](https://en.wikipedia.org/wiki/Algorithm) a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given [problem.](https://en.wikipedia.org/wiki/Problem_solving) Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

iii. Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the” flow” of data through an information system. A data flow diagram can also be used for the visualization of data processing(Structural design).It is common practice for a designer to draw a context level DFD first which show the interaction between the system and outside entities the context-level DFD is the ”exploded” to show more details of the system being modelled.

The DFD showing the top level of the system is called “Context Diagram”. It should be overview including basic inputs, processes and outputs. Then it is exploded in to more detailed lower level diagram that shows additional features of the system.

The purpose of DFD is to provide a semantic bridge between users and system developers. The diagrams are graphical, eliminating thousands of words, logical representations, modelling what system does; hierarchical, showing system at any level of details; and Jargon less, allowing user interaction and reviewing.

The goal of data flow diagramming is to have a commonly understood model of a system. The diagram is the basis of structured system analysis. The Data flow diagram, also known as “Bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become program in system design. The bubble represents the data transformations and the lines represent data flows in the system.

**5.1.1.1 BASIC DFD SYMBOLS**

* **Rectangles** - representing external entities, which are sources or destinations of data.
* **Arrows** - representing the data flows, this can either be electronic data or physicalitems. It shows the directional movement of data to and from External Entities, the process and Data Stores.
* **Open-ended rectangles** or **two parallel lines** –representing data stores, includingelectronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper.
* **Circle or a Rounded Rectangle-** representing processes, which take data as input, dosomething to it, and output it. It is used to represent functions.
  + - 1. **COMPONENTS OF DATA FLOW DIAGRAM**

**Source or Destination of Data :**

**Data Flow :**

**Data Processing :**

**Data Storage :**

Figure 5.1: DFD components

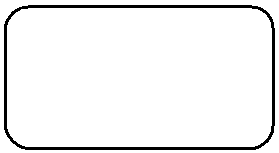
**5.1.1** **Activity Diagram and Flow Chart**

i. Activity Diagram

An activity diagram is a UML behavior diagram that represents the workflow of stepwise activities of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behavior diagrams. An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram is used by developers to understand the flow of programs on a high level.

Components of Activity Diagram

Start Point/initial state

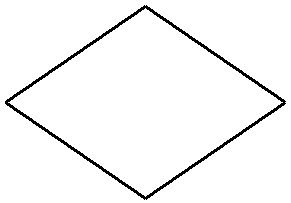


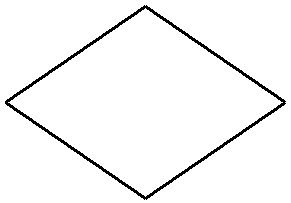
Activity

Action flow

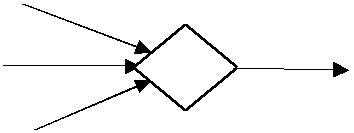


Class/object





Decision/branching



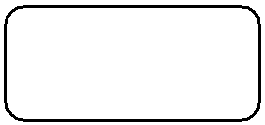
Merge

Fig 5.1.1 Components of activity diagram

ii. Flowchart

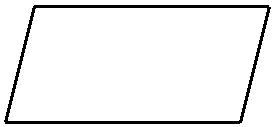
A flowchart is a type of [diagram](https://en.wikipedia.org/wiki/Diagram) that represents a [workflow](https://en.wikipedia.org/wiki/Workflow) or [process.](https://en.wikipedia.org/wiki/Process) A flowchart can also be defined as a diagrammatic representation of an [algorithm,](https://en.wikipedia.org/wiki/Algorithm) a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given [problem.](https://en.wikipedia.org/wiki/Problem_solving) Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

* Components of Flowchart



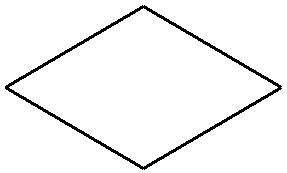
Start/End

Connector



Input/output

Process



Decision

Figure 5.1.2 Components of flow chart

* + 1. **PROJECT DATA FLOW DIAGRAM/ UML**

**BLOCK DIAGRAM & EXPLANATION**

**5V POWER**

**Alcohol Sensor**

**(MQ3)**

**Bluetooth Module**

**Arduino Nano**

**ANDROID APPLICATION**

**FILE**

**GRAPH**

--------------------------------------DATA-----------------------------------------------0000000000

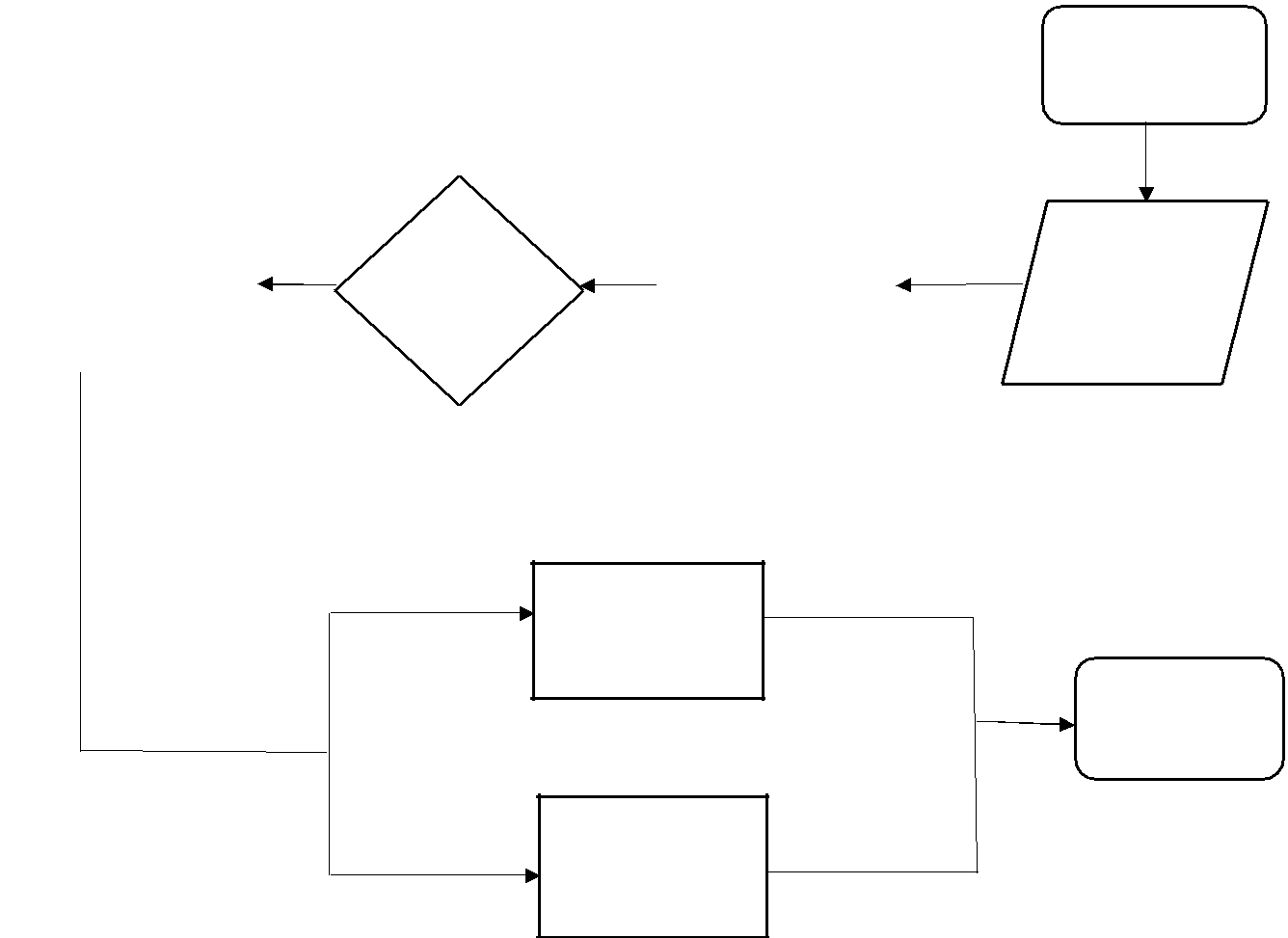
GPS

TIME

PHOTO

In this system we use Arduino nano board powered with 5V battery attached to MQ3 Alcohol Sensor .Here the input to the system is the breath of the vehicle driver. The Mq3 Alcohol Sensor is used for getting the analogue data about the amount of alcohol present in the persons blood and other related data such as type of alcohol which was consumed is also obtain using this sensor. Then this data is send to the smart phone using Bluetooth module then the smart phones android app will process this data and make instant capturing of the image of the driver as well as collection of his data such as license number, vehicle number and location details. Then a pdf file is generated which contains all these details and a copy of this details is stored in the cloud for obtaining the history as well as for further verification such as the same person is caught before or not if yes then we can double the penalization.

ii. Fig 5.1.2.2 Flow Chart- Driver Alcohol Detection



Start

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | If pin |  | High | MQ-3 |  |
|  |  |  |  | Arduino Nano |  |
|  |  |  |  |  |  |
|  | Alcohol |  |  |  |  |
|  |  |  |  | alcohol |  |
|  |  |  | is high |  |  |
|  | Detection |  |  |  | Low | sensor |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | Alert and | | |  |  |  |  |  |
|  |  |  |  |  |  |
|  | Information | | |  |  |  |  |  |

Vehicle Owner

Stop

Capture Image

Collect Details

Store Data

### Use Case Diagram

Breath Captured By Alcohol Sensor

Check Breath alcohol Percentage

If Value Above legal limit

Capture Person Picture, Obtain details

Store Data In Database

Figure 5.3: Use case diagram

**5.2 DATABASE DESIGN**

A database is a collection of interrelated data stored with minimum redundancy to serve users more quickly and efficiently. The general objective of a database is to make information access easy, quick, inexpensive, integrated and shared by different applications and users. Database design is an important yet sometimes overlooked part of the application development lifecycle. An accurate and up-to-date data model can serve as an important reference tool for Database Administrators, developers, and other members of joint application development team. The process of creating a data model helps the team uncover additional questions to ask of end users. Effective database design also allows the team to develop applications that perform well from the beginning. By building quality into the project, the team reduces the overall time it takes to complete the project, which in turn reduces project development costs. The central theme behind database design is to "measure twice, cut once". Effective database designers will keep in mind the principles of normalization while they design a database.

**5.3 INPUT DESIGN**

Input designing is the basic theory to be considered during system study. The input media used in the system is the keyboard. Details are entered in the system through different data entry screens. The system is designed in a user-friendly manner. Appropriate error messages are displayed when a false data is entered. Design of the system is web-oriented and is highly interactive to the users. The user interface design is very important for any application. The interface design defines how the software communicates within itself, to system that interpreted with it and with human who use it. The interface design is very good; the user will fall into an interactive software application.

The input design is the process of converting the user-oriented description of inputs into a programmer-oriented specification. The objective of input design is to create an input layout that is easy to follow and prevents the user from committing errors. It covers all phases of input, right from the creation of initial databases to the actual data entry into the system. The input design is the link that ties the system into the world of its users. Hence, lays its importance in the design phase. The input design makes sure that while entering data, the end-users understand the format in which the data is to be entered so that it is accepted by the system, the data values that are mandatory for the system to function, the order in which transactions need to be processed etc.

The goal designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection feature and adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right message and help for the user at right time are also considered for the development of this project. Input design, involves determining the record media, method of input, speed of capture and entry to the system.

* **Hardware inputs are:**

 Camera : Image Processing and Feature Extraction.

* Alcohol Sensor and Camera : Detection of driver’s alcohol intoxication and capture the driver image.
* **Input form:**

 Login page : System administrator can activate the system with

Username, Password and Vehicle number.

**5.4 OUTPUT DESIGN**

Output is the most important one to the user. A major form of the output is the display of the information gathered by the system and the servicing the user requests to the system. Output generally refers to the results or information that is generated by the system. It can be in the form of operational documents and reports. Since some of the users of the system may not operate the system, but merely use the output from the system to aid them in decision-making, much importance is given to the output design.

Output generation hence serves two main purposes, providing proper communication of information to the users and providing data in a form suited for permanent storage to be used later on. The output design phase consists of two stages, output definition and output specification. Output definition takes into account the type of outputs, its contents, formats, its frequency and its volume. The output specification describes each type of output in detail.

The objective of the output design to covey the information of all the past activities, current status and emphasize important a quality output is one, which meets the requirements of the end user and presents the information clearly.

**5.5 PROGRAM DESIGN**

**CHAPTER 6**

**FUNCTIONAL AND NON-FUNCTIONAL**

**REQUIREMENTS**

**6.1 FUNCTIONAL REQUIREMENTS**

In software engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Generally, functional requirements are expressed in the form “system must do requirement”. Functional requirements for each of the uses cases described below:

* Descriptions of data to be entered into the system.
* Descriptions of operations performed by each input.
* Descriptions of work-flows performed by the system.
* Descriptions of system outputs.
* How the system meets applicable regulatory requirements.

**6.2 NON-FUNCTIONAL REQUIREMENTS**

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviour’s. Non-functional requirements are “system shall be requirement ". Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes”, “quality goals", "quality of service requirements" and "non-behavioural requirements. Some of the non-functional requirements are mentioned below:

* **Performance requirements**

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance.

* **Operating constraints**

List any run-time constraints. This could include system resources, people, needed software, etc.

* **Platform constraints**

Discuss the target platform. Be as specific or general as the user requires. If the user doesn't care, there are still platform constraints.

* **Accuracy and Precision**

Requirements about the accuracy and precision of the data. Beware of 100% requirements; they often cost too much.

* **Modifiability**

Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person- months).

* **Portability**

The effort required to move the software to a different target platform. The

measurement is most commonly person-months or % of modules that need changing.

* **Reliability**

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures). The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

* **Security**

One or more requirements about protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, time) to break into the system. Do not discuss solutions (e.g. passwords) in a requirements document.

* **Usability**

Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

**CHAPTER 7**

**TESTING**

**7.1 TESTING STRATEGIES**

An engineered product can be tested in one of the two ways. These testing strategies include:

* **Black box Testing**
* **White box Testing**

These testing strategy checks the correctness of every statement in the program and results in execution of every instruction in the program module.

**Black box testing:**

Knowing the specified function that a product has been designed to perform, test can be conducted that each function is fully operational. Black box test is carried out to test that input function is properly accepted and output 3 correctly produced. This test examines some aspects of system with little regard for the internal structure of the software.

In our project we use black box testing, we have no clear idea about the coding and we just test the external structure. That is we just try to test that the correct output will come based on the input we give.

Errors found through black box testing are:

1. Incorrect or missing function
2. Interface errors
3. Errors in database structure or external database access.
4. Performance errors.

v. Initialization and termination errors.

**2. White box testing**

White box test of software is predicted on a close examination of procedural detail. The status of the project may be tested at various points to determine whether the expected or asserted status is corresponding to the actual status. We use white box testing, because in this testing all the internal functions and operations are tested. In our project, we have three modules. We separately test these three modules and its internal structure.

Using these following test cases can be derived:

i. Exercise all logical conditions on their true or false side.

ii. Exercise all loops within their boundaries and their operation bounds.

**7.2 UNIT TESTING**

Unit testing focuses on verification effort on the smallest limit of software design. Using the unit test plan prepared in the design phase of the system, important control paths are tested to uncover the errors within the module. This testing was carried out during the coding itself. In this testing step each module is going to be working satisfactorily as the expected output from the module.