SMART POLLUTION UNDER CONTROL APPLICATION

**Abstract:**

Smart pollution under control application (SPUC) is built to control the emission of poisonous gases like CO, SO2, hydrocarbons etc. from the vehicles and to eliminate the use of PUC or RTA vehicles for checking the air pollution level in the vehicles. In this project we have considered emission of CO percent (ppm) which can be detected by the various digital sensors (presumably to be installed in every vehicle) that are available in the market (for example MQ7). The values that are obtained from the sensor are stored in the file (for now, in the prototype) and are read in the application. If the emitted value is greater than a set permissible threshold, then the alert message is sent to the user to get the vehicle serviced in the nearest RTO office before a given deadline. In case of no response from the user, notice of a fine of Rs.1000 (or whatever the government seems fine) is sent to the user’s address along with a notification in the application.

**Concept:**

There is a lot of pollution in this country. Presently, we are using vehicles (PCU Vans) to check the pollution values of other vehicles! Isn’t that sort of an irony? So, we thought of coming up with a Smart Idea. Our idea would use digital CO detectors, which would send values (in ppm) to our android application. The application will continuously update a central database hosted by the RTO (developed by us). As soon as the database encounters a value for a vehicle where CO levels have crossed the threshold value, a notification will be sent automatically to that vehicle’s owner along with a deadline date. Also, if the owner doesn’t get his vehicle serviced, a notice of a fine will be sent to his address. We came up with one more thought. Presently, a traffic policeman catches a vehicle and checks its PUC Sticker. But now, he can just ask the owner to show the status of his Pollution Emissions in the application. For viewing the statistics and other details in the central database, we have developed a Web Page which displays them.

**Project:**

Every vehicle should be Registered once using their vehicle ID and the ID of the sensor that is attached to the particular Vehicle. Registration details contain Vehicle ID, Sensor ID, Name of the vehicle owner, Phone number, Address (For security purposes manual address verification can be done). Once the vehicle has been registered you can just sign in or sign out using the vehicle Id and the sensor id.

CO level is obtained whenever the vehicle’s engine has been turned on and turned off. We don’t need to carry our license papers along with us we can also save them in the profile slot, if in case we need to show them to the traffic police.

The project involves creation of a central user database where all these data is stored. All this information will be at the RTO office to manage for a given city. All query privileges like checking CO level, setting a deadline if CO threshold has been crossed by a vehicle and giving a notice of fine lies with admin of this database which can be the RTO office.

**Implementation:**

Sensor

Text file

CO values

Database (back end) end

Mobile App

6000(ppm)

Current Co level

Alerts:

Your co level has exceeded the threshold level Please report at the nearest RTO office before (deadline)

Vehicle id, sensor id, owner name, address, phone number, co level ,deadline

Updates the

Current co level

Of the vehicle

If crosses the threshold

Alert along with a

Deadline is sent to the

User.

For the view of the

RTO officer (web)

* List of all the vehicles that are registered
* List of vehicles that have crossed the deadline
* List of vehicles that have crossed the threshold value.

Fig 1.1Working prototype of the application

The mobile application is built in android and the backend database is in PHP and MySQL. The figure 1.1 is the working prototype of our application

Sensor sends the current value to the text file in the application which is displayed on the user’s screen (front end of the app) and the value is updated in the database that has all the registered vehicles’ ID, sensor id, address, name of the owner, phone number, current CO value. Once the value reaches the threshold and ALERT message is sent to the user of vehicle along with the deadline, before which his vehicle needs to report at the RTO office to get checked.

For the **officer** to view the list of all the vehicles that have reached the threshold, which have crossed the deadline, all that are registered etc. can be seen on the web page of the website.

**VIEWS:**

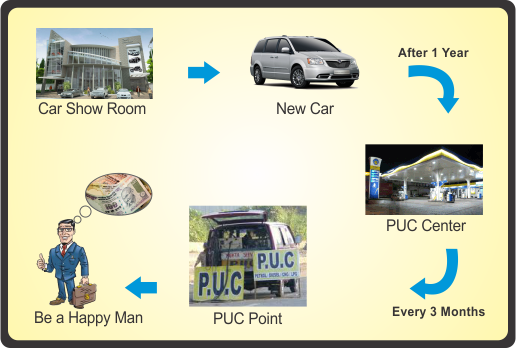
**Future extensions of the Project:**

1. This project can be further extended to all gases not only CO, using a different kind of sensor. By this we can control pollution to a large extent SMARTLY!
2. This app can also be synchronized with your Gmail or whatsapp for the alert messages. (Also, we have thought of building a Whatsapp Bot. this can phase out the issue that an extra app for SPUC is required in each android phone. Whatsapp users can just download our bot by just a saving our bot’s contact number details in an android phone. Then, in the whatsapp chat with the bot, an owner can use functions like #status for present status of alerts or deadlines, #profile for viewing his profile details, #contactus for contacting the RTO office, and many more).
3. We can extend it to have vehicle security through the sensor tracking and route finder.

**Cost analysis:**

1. Sensor cost and its maintenance costs.
2. Database maintenance.

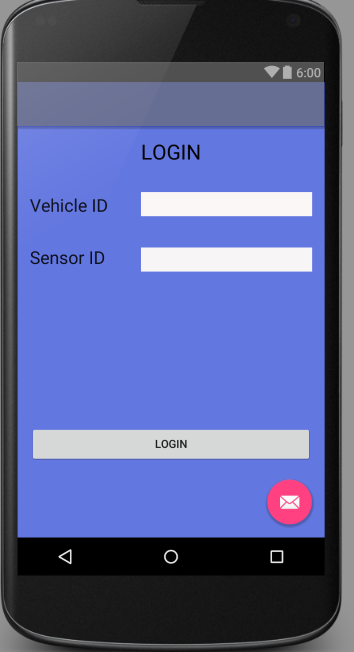
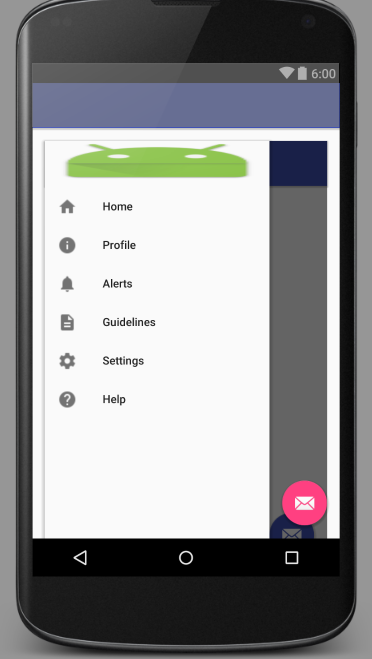
This is a cheaper way compared to the Pollution under control vehicles and the employment needed to maintain these vehicles(PUC).



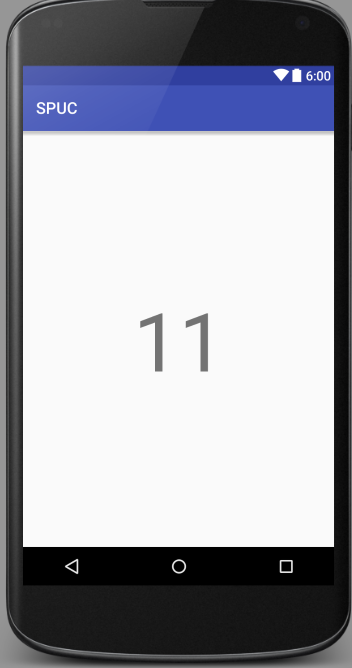
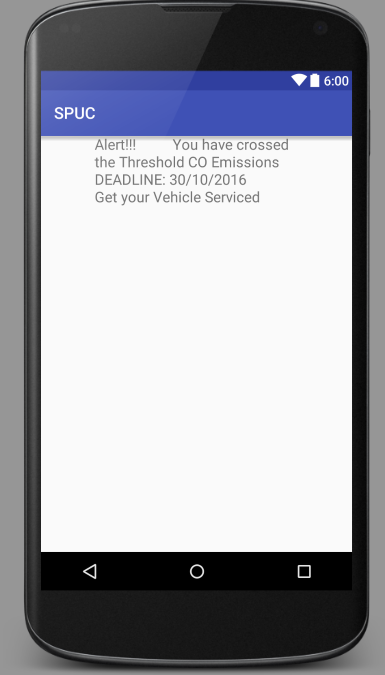
Instead of this just download the app to check your CO level

**Some Screenshots from our application:**

**Login Page Home Page**

** **

**Alert Message Current CO Emission**



**THANK YOU**

**TEAM:**

1. **Amrutha Nanduri.**
2. **Nikunj Gupta.**
3. **Vijaya Sindhu.**
4. **Yadlapalli Abhijnu.**

**Resources**:

1.https://en.wikipedia.org/wiki/Carbon\_monoxide\_detector

2. http://www.motortraffic.gov.lk/web/web/index.php?option=com\_content&view=article&id=123&Itemid=149&lang=en

3. http://www.detectcarbonmonoxide.com/co-health-risks/

4. http://forum.arduino.cc/index.php?topic=268069.0