**IOT**​ **MINIPROJECT REPORT**

**ON**​

​**COLOR SENSING FOR BLIND**

​**Submitted By: Dhruv Chaudhary(09)**

**Vedhas Kharche(27)**

**Aniket Neema(42)**

**Shubham Kothawade(29)**

**Rishabh Bagul(03)**

**Branch : T. E. Computer (2018-2019)**

**Guided By:** ​​**prof.I.Priyadarshini**

​**Department of Computer Engineering**

**K. K. Wagh Institute of Engineering Education & Research Hirabai Haridas Vidyanagari, Amrutdham, Panchavati, Nashik – 422 003.**

**K. K. WAGH INSTITUTE OF ENGINEERING EDUCATION &**

**RESEARCH NASIK.**

**Affiliated to Savitribai Phule Pune University**

***CERTIFICATE***

This is to certify that

**Dhruv Chaudhary(09)**

**Vedhas**​ **Kharche(27)**

**Aniket Neema(42)**

**Shubham Kothawade(29)**

**Rishabh Bagul(03)**

Has Successfully completed

IOT Mini Project on

**COLOR SENSING FOR BLIND**

**prof. I. Priyadarshini**

**Prof.Dr. S. S.Sane**

**Prof. Dr. K. N. Nandurkar**

[Mini Project Guide]

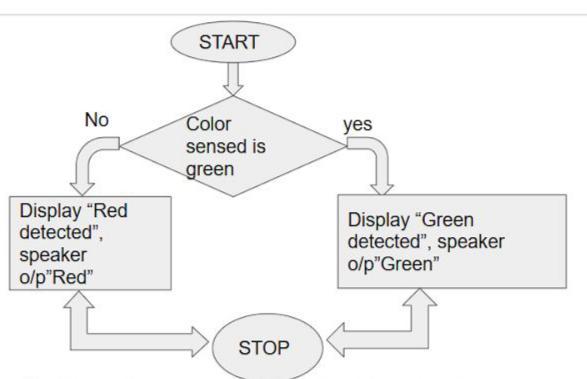
[H.O.D]

[Principal]

**Step 1: Purpose and Requirement Specification**

* **Purpose : With the help of color sensor and speaker, the blind can get correct information and thus he can act accordingly.**
* **Behavior : The system will check the ranges of color sensed and generate a message of maximum sensed color on speaker**
* **System Management Requirement : The system provides facility to convert digital data into sound format using libraries.The system also provides other control signals as well**
* **Data Analysis Requirement : Data Analysis is based on local data generated by color sensor.**
* **Application Deployment Requirement : Application is installed on a local device and is locally accesible(no cloud involved)**

**Step 2: Process Specification**



There are two ways for detection:

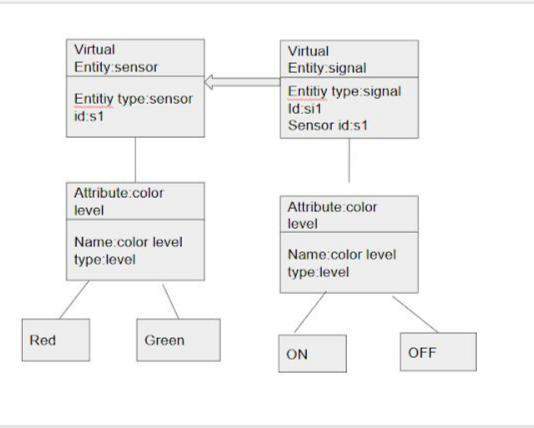
1. When signal is Green
2. When signal is Red

Both ways involves same action of displaying message on serial monitor and passing sound to blind about the respective color,that is if color is red output will be “Red” on serial monitor and sound.

**Step 3: Domain Model Specification**

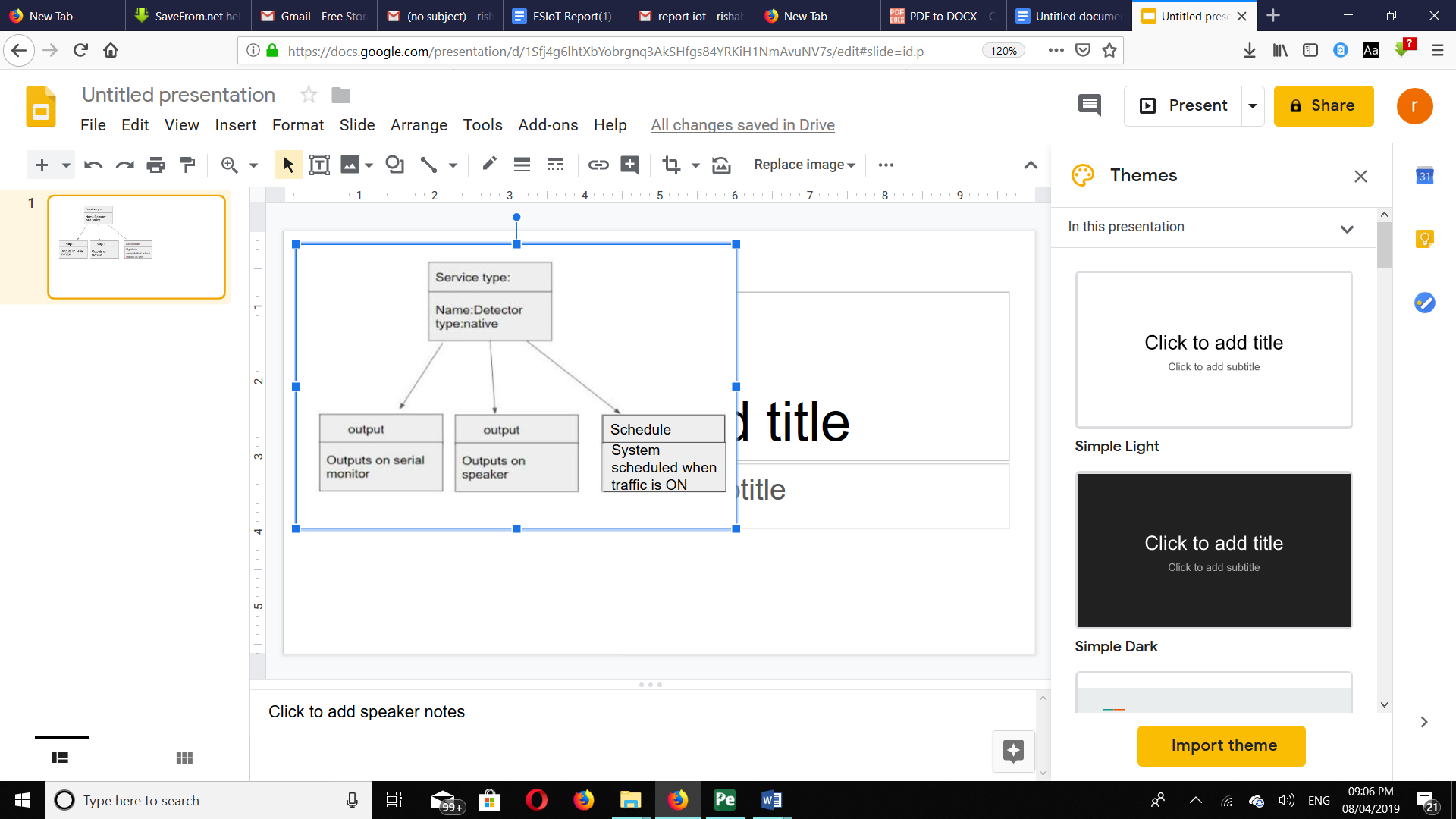
* **Physical Entity :** ​signal,sensor
* **Virtual Entity :** signal,sensor
* **Device :** ​Color Sensors,Amplifier,speaker connected to Arduino
* **Resource :** ​Operating system which runs on Arduino Uno boardand memory support
* **Service :** ​Detecting color.

**Step 4: Information Model Specification**



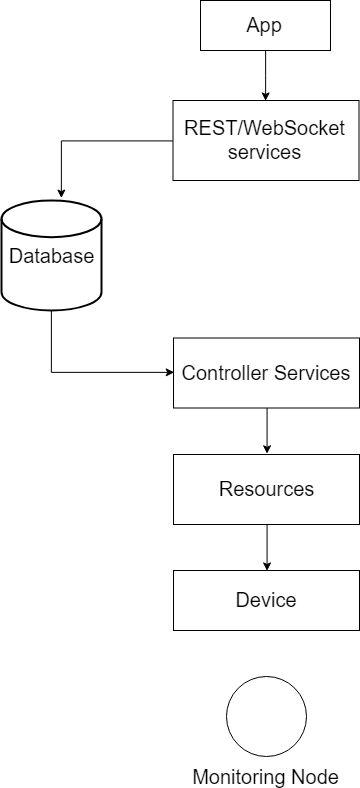
This IOT Information model maintains the necessary information about virtual entities and their attributes. When the signal is ON,it is sent to sensor , which depending on each color’s intensity decides whether the color is green or red.When signal is off,no sensing is being done.system is in sleep state.

**Step 5: Service Specifications**



Service specification defines list of services provided by the system. It includes service type, input/output, etc. Here only service provided is detecting.Output of service include outputing on both serial monitor and to the speaker as well the respective colour name(“Red” when traffic light is red)and scheduled when the traffic light is ON the next time.

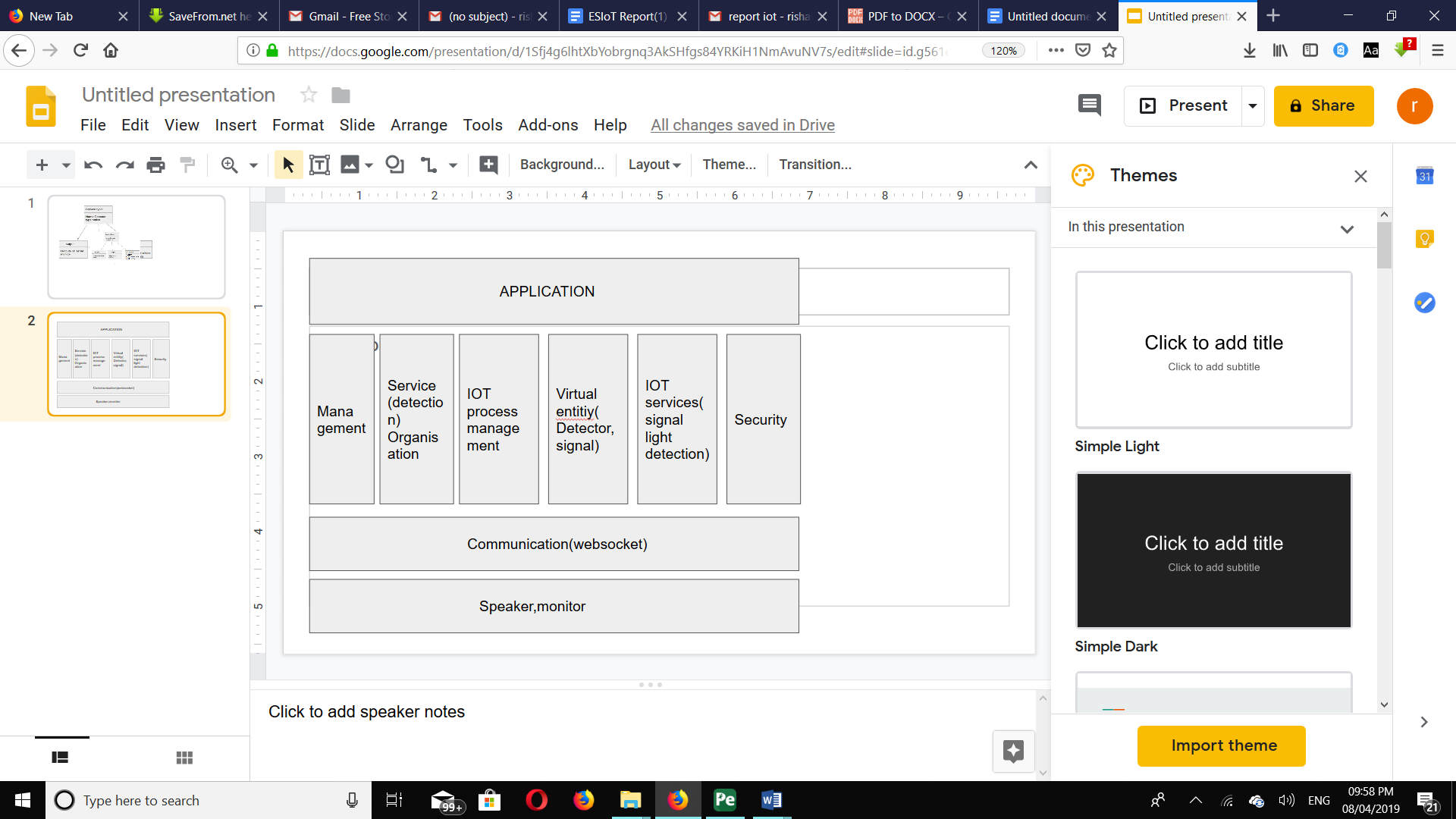
**Step 6: IoT Level Specification**



Our project is local detect,local anaylsis modelled one.No cloud is involved so it lies in level 1 of IOT.Here communication with the database is made using websocket protocols and controller services(detector) manage to flow the service(the display of respective color) to the devices(speaker,screen).

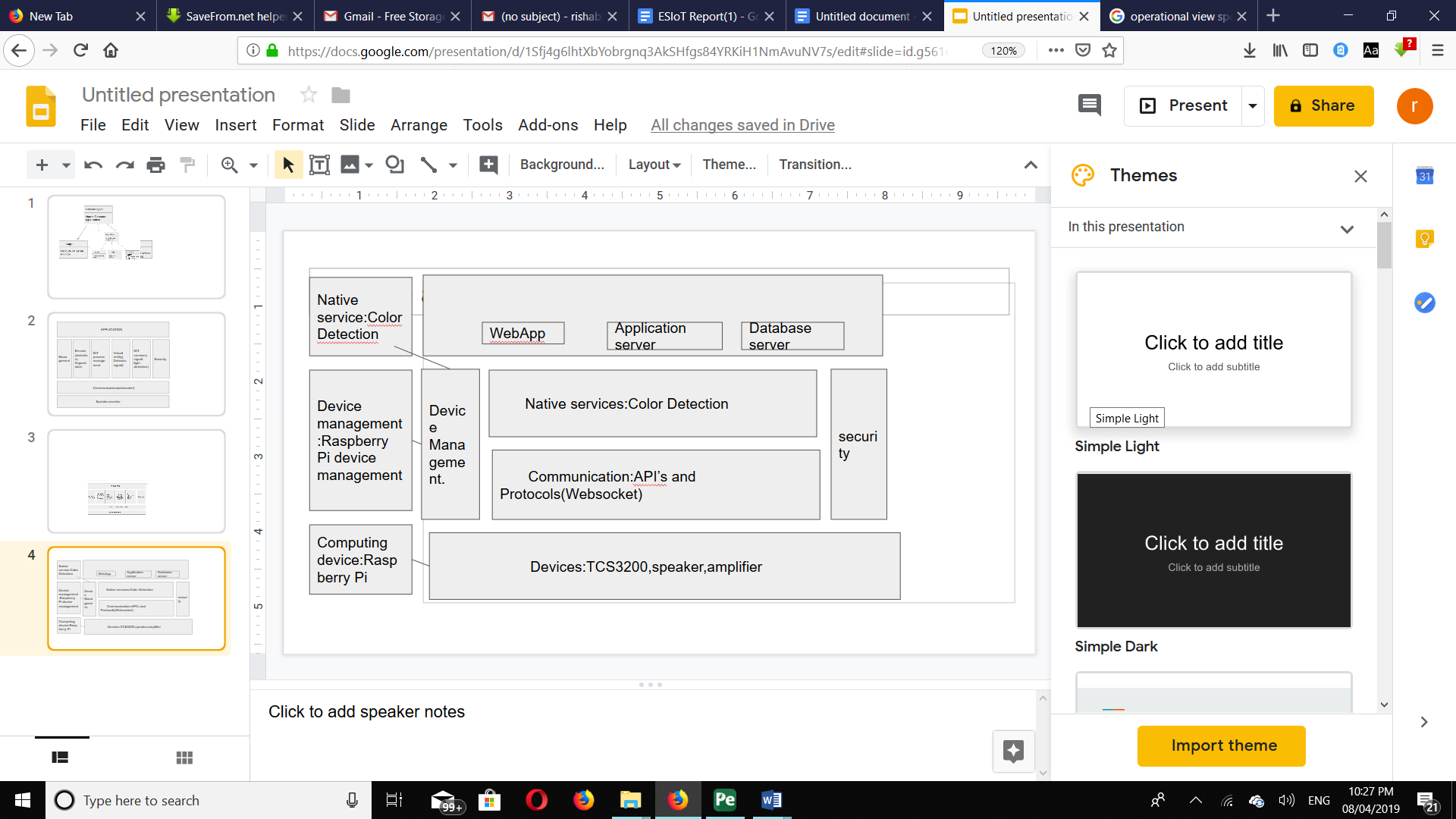
**Step 7: Functional View Specification**

Above model describes the perspective of the system. Functional view describes the system’s runtime Functional components, their responsibilities, default functions, interfaces, and primary interactions.



Security as such is not so important for our model as its operating in local mode.Starting from the bottom, Here devices include the speaker and the amplifier.Communication is done using websocket protocol as its level 1 model.Services include detection and displaying information for the blind.Virtual entity include signal and sensor.

**Step 8: Operational View Specification**



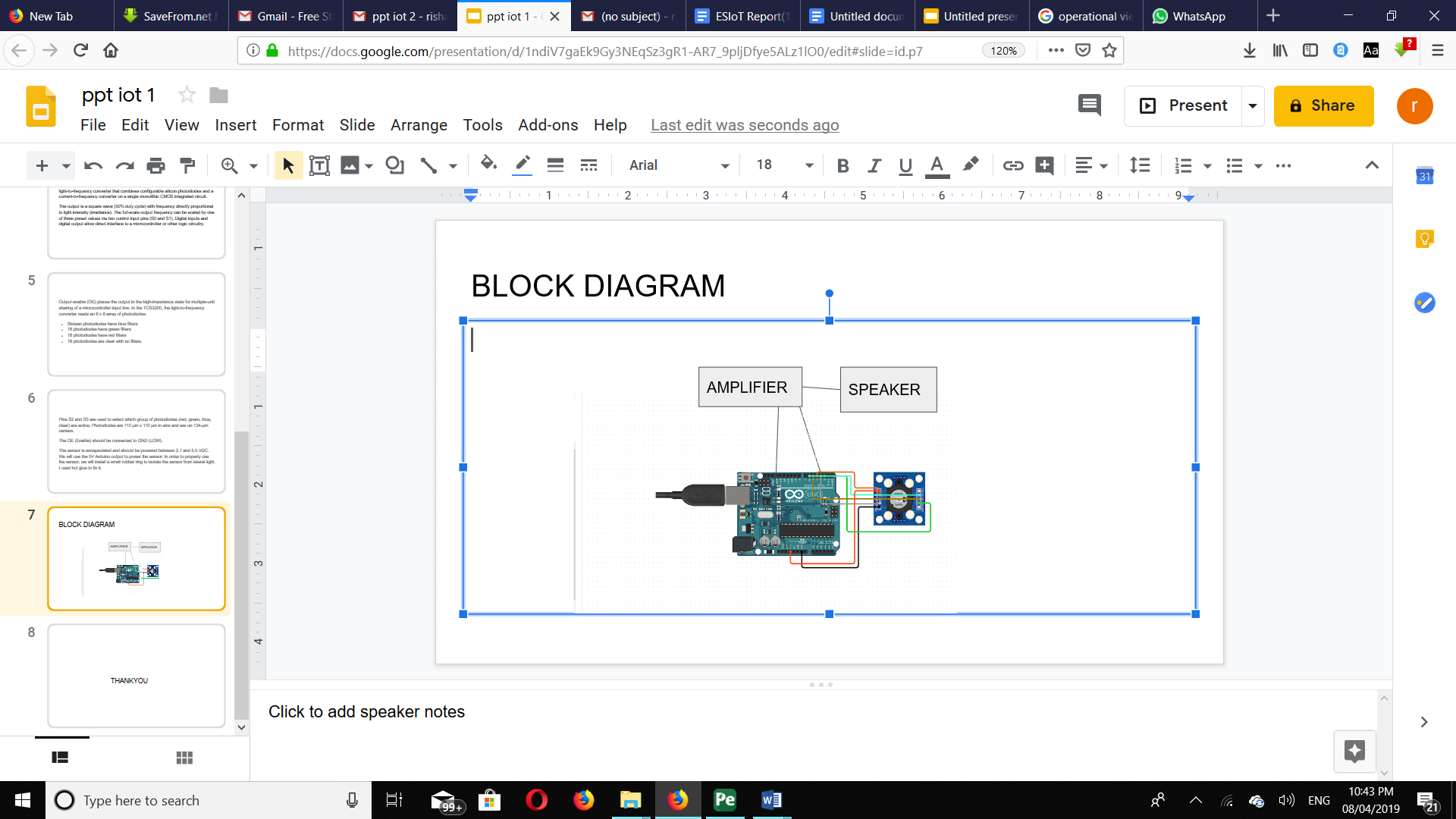
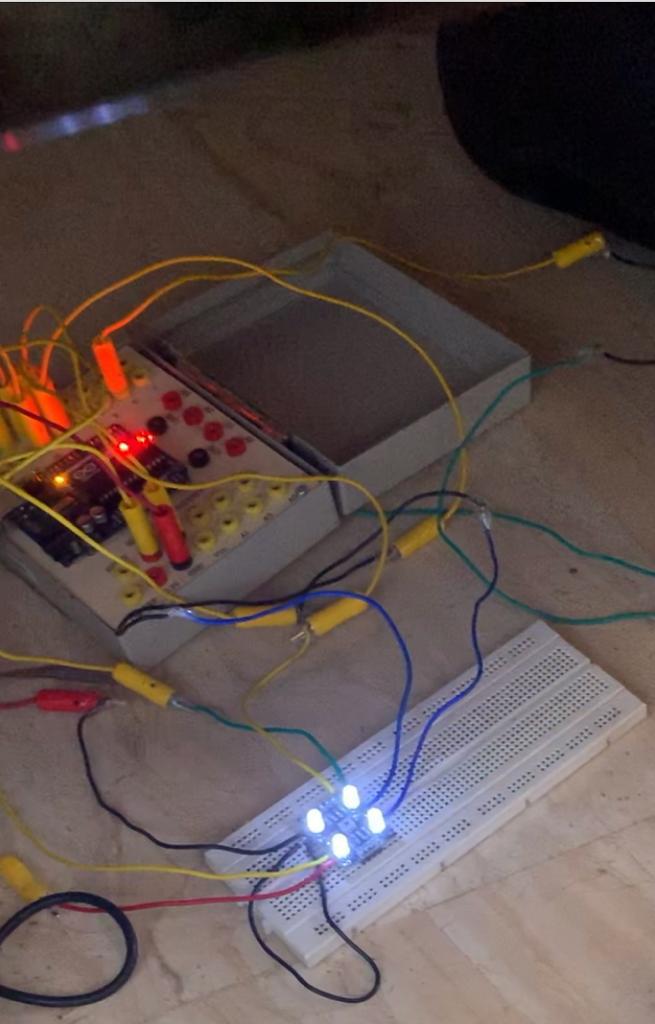
Operational view depends on the specific actual use case and requirements. Smart objects in IOT uses different methods for communication using different technology. This is used to address how actual system can be realized by selecting technologies and making them communicate and operate in comprehensive way.

**Step 9: Device and Component Integration**

The devices and components used for this projects are: Arduino Uno board,TCS3200 color sensor,mini speakers,ampifier, Battery.

TCS3200 is directly connected to Arduino UNO board. Amplifier and speaker is connected to UNO using bread board

color sensor is kept in alignment with traffic signal so it will read the data and speaker can be connected via headphones for sound delivery.



**Step 10: Application Development**

A blind will be able to sense the traffic signal and not just the system senses the light but gives proper message to blind to cross the road or not.The light can be converted to sound waves by mapping functions given in talkie library.

**Source code:**

**// TCS230 or TCS3200 pins wiring to Arduino**

**#include "talkie.h"**

**#define S0 8**

**#define S1 9**

**#define S2 10**

**#define S3 11**

**#define sensorOut 12**

**Talkie voice;**

**const uint8\_t spRED[] PROGMEM = {0x6A, 0xB5, 0xD9, 0x25, 0x4A, 0xE5, 0xDB, 0xC5, 0x4F, 0x6D, 0x88, 0x95, 0x2D, 0xD2, 0xB4, 0x8F, 0x2E, 0x37, 0x0E, 0x33, 0xCF, 0x7E, 0xAA, 0x9A, 0x5C, 0xC3, 0xB4, 0xCB, 0xA9, 0x86, 0x69, 0x76, 0xD3, 0x37, 0xB7, 0xBE, 0xCD, 0xED, 0xEF, 0xB4, 0xB7, 0xB0, 0x35, 0x69, 0x94, 0x22, 0x6D, 0x10, 0x28, 0x42, 0xB9, 0x8B, 0xC8, 0x06, 0x00, 0x50, 0xCF, 0x0E, 0xEE, 0x62, 0xEA, 0xA6, 0xBC, 0xC3, 0x14, 0xBB, 0x4A, 0x9F, 0xFA, 0xA5, 0xAF, 0x25, 0x13, 0x17, 0xDF, 0x9C, 0xBF, 0xFF, 0x07};**

**const uint8\_t spGREEN[] PROGMEM = {0x64,0xD5,0xA2,0x22,0x23,0xAC,0xB0,0x4D,0xF1,0xCA,0x2C,0x55,0x1A,0xF6,0x6C,0x3F,0x24,0xC4,0x72,0x19,0xB2,0xCA,0xA0,0x62,0x67,0xAD,0x8B,0x49,0xCD,0x53,0xDC,0x8D,0x3A,0x55,0x0E,0x4D,0xB5,0x3F,0xAA,0xD2,0x38,0x5C,0xBD,0xFD,0xAA,0x5A,0x51,0x76,0xB7,0x2D,0xA3,0xEE,0xC5,0xD1,0xD2,0x6F,0xAD,0x66,0x78,0x43,0xDB,0x28,0x35,0xDA,0x61,0x15,0xED,0x22,0x4C,0x6B,0x87,0x15,0x8A,0x73,0xB3,0xAD,0xEB,0x52,0xB9,0x4E,0xAD,0xB6,0xAE,0x29,0xB2,0x09,0x0B,0x5B,0xDA,0xA6,0xB1,0xCA,0xDC,0x69,0x69,0xAB,0xC2,0x0A,0x73,0xD5,0xA5,0x6D,0x9A,0xD5,0x3D,0x12,0x95,0xB6,0x7A,0x15,0xB7,0x8A,0x58,0xDA,0xE2,0x54,0x32,0x42,0x62,0x69,0x8A,0x13,0x35,0xCD,0x48,0xFF,0x0F};**

**// Stores frequency read by the photodiodes**

**int redFrequency = 0;**

**int greenFrequency = 0;**

**int blueFrequency = 0;**

**// Stores the red. green and blue colors**

**int redColor = 0;**

**int greenColor = 0;**

**int blueColor = 0;**

**void setup() {**

**// Setting the outputs**

**pinMode(S0, OUTPUT);**

**pinMode(S1, OUTPUT);**

**pinMode(S2, OUTPUT);**

**pinMode(S3, OUTPUT);**

**// Setting the sensorOut as an input**

**pinMode(sensorOut, INPUT);**

**// Setting frequency scaling to 20%**

**digitalWrite(S0,HIGH);**

**digitalWrite(S1,HIGH);**

**// Begins serial communication**

**Serial.begin(9600);**

**}**

**void loop() {**

**// Setting RED (R) filtered photodiodes to be read**

**digitalWrite(S2,LOW);**

**digitalWrite(S3,LOW);**

**// Reading the output frequency**

**redFrequency = pulseIn(sensorOut, LOW);**

**redColor = map(redFrequency, 70, 120, 255,0);**

**// Printing the RED (R) value**

**Serial.print("R = ");**

**Serial.print(redColor);**

**delay(100);**

**// Setting GREEN (G) filtered photodiodes to be read**

**digitalWrite(S2,HIGH);**

**digitalWrite(S3,HIGH);**

**// Reading the output frequency**

**greenFrequency = pulseIn(sensorOut, LOW);**

**greenColor = map(greenFrequency, 100, 199, 255, 0);**

**// Printing the GREEN (G) value**

**Serial.print(" G = ");**

**Serial.print(greenColor);**

**delay(100);**

**// Setting BLUE (B) filtered photodiodes to be read**

**digitalWrite(S2,LOW);**

**digitalWrite(S3,HIGH);**

**// Reading the output frequency**

**blueFrequency = pulseIn(sensorOut, LOW);**

**blueColor = map(blueFrequency, 38,84 , 255, 0);**

**// Printing the BLUE (B) value**

**Serial.print(" B = ");**

**Serial.print(blueColor);**

**delay(100);**

**// Checks the current detected color and prints**

**// a message in the serial monitor**

**if(redColor > greenColor && redColor > blueColor){**

**Serial.println(" - RED detected!");**

**voice.say(spRED);**

**delay(100);**

**}**

**if(greenColor > redColor && greenColor > blueColor){**

**Serial.println(" - GREEN detected!");**

**voice.say(spGREEN);**

**delay(100);**

**}**

**if(blueColor > redColor && blueColor > greenColor){**

**Serial.println(" - BLUE detected!");**

**}**

**}**

|  |  |  |  |
| --- | --- | --- | --- |
| **Report Documentation & Accounting Page** | | | |
| Report Code: CS-TE-IOT Mini Project 2018-2019 | | **Report Number: TE-B 03,09,27,29,42** | |
| **Address (Details):**  **Computer Department, K. K. Wagh Institute of Engineering Education & Research,**  **Hirabai Haridas Vidyanagari, Amrutdham, Nashik**  **Pin – 422 003** | | | |
| Report Title: Color sensing for blind | | | |
| **Author [phone, E-mail]:**  **phone- 7743834321**  **E-mail : rishabhbagul3@gmail.com** | | **Author Details (Year, Branch, Roll):**  **Year:** 2018– 2019  **Branch:** Computer Engineering  **Roll: 03,09,27,29,42** | |
| Type Of Report: FINAL | Time Covered (From – To)  **16-02-2019**  TO  **9-04-2019** | **DateOf Report**  (DD-MM-YYYY)  **9-04–2019** | **Page Count**  17 |
| ***Key Words*: TCS3200,traffic,amplifier.** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Report Checked By: | Report Checked Date: | **Guides Complete Name:**  **Prof.I.Priyadarshini** | Total Copies  **1** |
| **Abstract:**  **NOTE –** ​**The mini project color detection for blind is an** **application which helps blind to detect the color of signal being shown using a color sensor TCS3200 and after sensing surrounding color can convert the message signal into speech using a library called “talkie” , which uses hexadecimal encoding to generate speech and speech being amplified using a 3 pin amplifier thus proving the purpose.** | | | |