**`**

**Product Design Document**

**Know Your Own Surrounding(KYOS)**

|  |  |
| --- | --- |
|  |  |

**By,**

**Group - 13**

Saloni Chudgar - 1641013

Manav Chotalia - 1641036

Dhruvi Gajjar - 1641038

Poojan Prajapati - 201501081

**CONTENTS**

**1** **INTRODUCTION 4**

1.1 Purpose of the Document 4

1.2 Intended Audience 4

**2** **SYSTEM OVERVIEW 5**

2.1 Top Level System Diagram 5

2.2 Description 5

**3** **SYSTEM ARCHITECTURE 6**

3.1 Architecture Diagram 6

3.2 Hardware and Communication Architecture 6

**4** **INTEGRATION DETAILS 8**

4.1 Integration Sequence 8

4.2 Integration Procedure 8

**5** **TOOLS FOR ARCHITECTURE AND DESIGN 9**

5.1 Software Tools 9

5.2 Hardware Tools 9

**6** **DETAILED DESIGN 10**

6.1 HMI / User Interface Design 10

6.2 Module 1 10

6.3 Module 2 10

**8** **DATA DESIGN 11**

8.1 Module 1 11

8.2 Module 2 11

**9** **CONSTRAINTS 12**

**10** **ASSUMPTIONS 13**

**11** **APPENDIX 14**

# INTRODUCTION

## Purpose of the Document

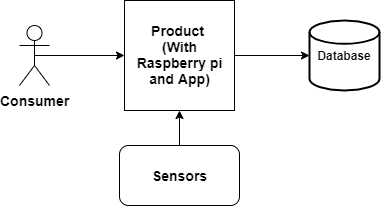
Project\_name is a wearable product which will allow the user to monitor Air-quality of his/her surroundings.The product is made for personal usage, which allows the user to customise threshold as per user’s requirements. Moreover, the device can be linked to cell phone via bluetooth or wifi. User can select any one for the connectivity. User will get an alert or update via mail and message.

## Intended Audience

This document is intended for developers, testers, the project team as well as the user.

# 1 SYSTEM OVERVIEW

## Top Level System Diagram



## Description

The product has the following features :

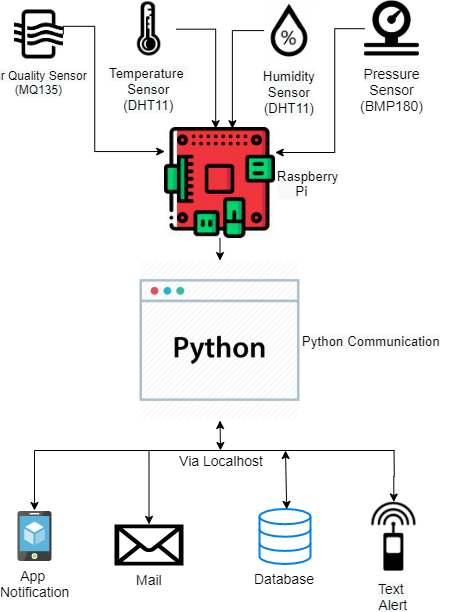
* It will display Air Quality, Temperature and Humidity.
* It will detect the level of individual components in the air and generate alert to mobile device or send mail to respected person in case of discrepancies.
* It will have warning systems for the prevention of undesired air pollution episodes for people with specific medical conditions.
* It facilitates Real Time Air Monitoring
* It can customize thresholding of concentration/s if and when required
* Remote Viewing is possible and the database can be remotely monitored
* Additional button for immediate measurement of air-quality at any given time
* Reset feature clears all the data.
* Automatic measurement of data through sensors every two minutes

The product must satisfy the following requirements :

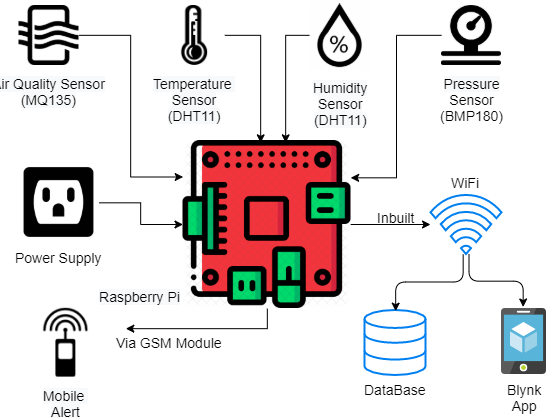
* The device allows user to keep a record of the Air Quality Parameters
* Alert is generated if Air Quality Parameters exceed certain levels
* Sensor data can be viewed by user
* Location must be updated as soon as network connectivity is available
* On reset, all data must be deleted
* Backup must be available for monitoring and analysis purpose

# SYSTEM ARCHITECTURE

## Architecture Diagram



## Hardware And Communication Architecture

**

# INTEGRATION DETAILS

## Integration Sequence

* Sensors Integration with Raspberry pi and data verification.
* Testing of combination of inputs from various sensors at the same time
* Sending/Receiving data from raspberry pi to Database.
* Mapping and transmitting/receiving data from raspberry pi to blynk.
* Testing of GPS module of blynk, receiving location via blynk to raspberry pi, Processing it and update blynk app.
* Send an alert or update mails, mobile notifications and text messages on the update.

## Integration Procedure

Sensor Integration:

Air Quality Sensor: The sensor senses air quality in milliseconds, sends data to raspberry pi in *string.* The *string* gives percentage of air quality goodness according to standard values of various gases set. This data is verified and processed by raspberry.

BMP180: Integrating this sensor to Air quality sensor. Testing the data similarly as we receive pressure measurement and calibrating according to the standards.

DHT11: Adding this module which senses humidity in percentage and temperature in celsius we tested three sensors and their processing timings.

Blynk: Configured blynk app with raspberry pi adding blynk library. Create token Blynk for new project, add that token to raspberry pi and thus connection. Test notification and alert generation from raspberry pi data to blynk.

Database Connection: PHPmyAdmin is connected to raspberry pi via local host. There by sending and receiving of data is tested. For security, Each user’s unique key is generated and encrypted. Furthermore code to store verified data for back-up, another replica is stored for each user for 3 months.

Testing: Complete testing of sending/ receiving data with test case and requirement traceability matrix is built for reference. Dependencies from testing sheet is checked. once tested deployment code is locked to avoid accidental change in code.

# TOOLS FOR ARCHITECTURE AND DESIGN

## Software Tools

* Blynk App/Blynk Tool/Blynk.io
* PhpMyAdmin - local host

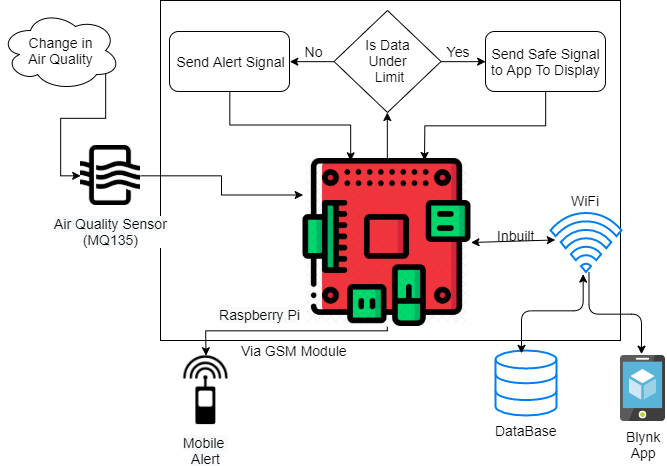
## Hardware Tools

● Raspberry pie

* Air Quality Sensors (MQ135)
* Temperature Sensor(DHT11)
* Humidity Sensor(DHT11)
* Pressure Sensor(BMP180)
* GSM Module
* Mobile GPS via-Blynk App
* Power Supply
* Switch for reset

# DETAILED DESIGN

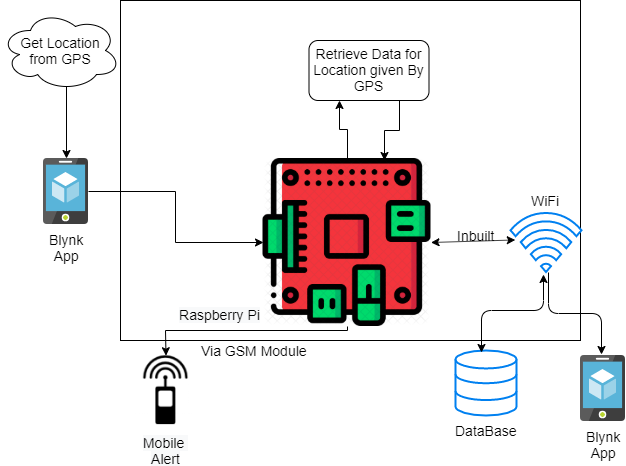
## Module 1



Description: Air Quality Sensor

In Module the data is being sent from Air Quality Sensor to the Raspberry Pi and than it is checked if the Data is under limit or not , if the data is under limit than a Safe Signal is sent to Mobile via GSM Module and Blynk App through Wifi from Raspberry Pi.If the data is over Limit than Alert Signal will be sent to the Mobile via GSM Module and to Blynk App via Wifi inbuilt on Raspberry Pi.

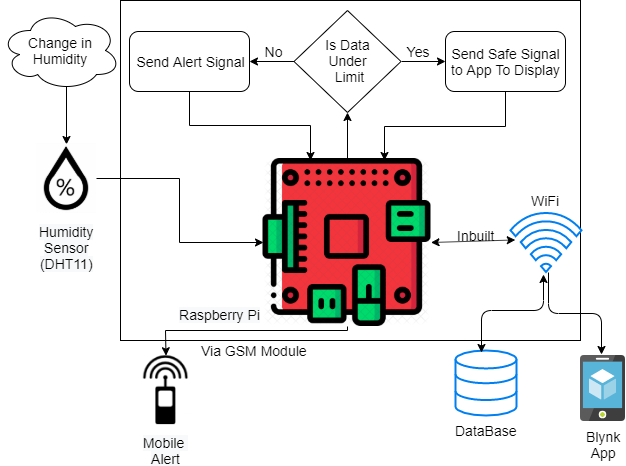
## Module 2



Description: GPS Location

GPS Location will be detected from BLYNK APP on the phone and would be sent to Raspberry Pi and compared to the data in the Database provided and if the Data is not under the Limit provided for that LONGITUDE and LATITUDE then an Alert Signal would be sent to Mobile via GSM module and same Alert would be sent to BLYNK App via Wifi of Raspberry Pi.

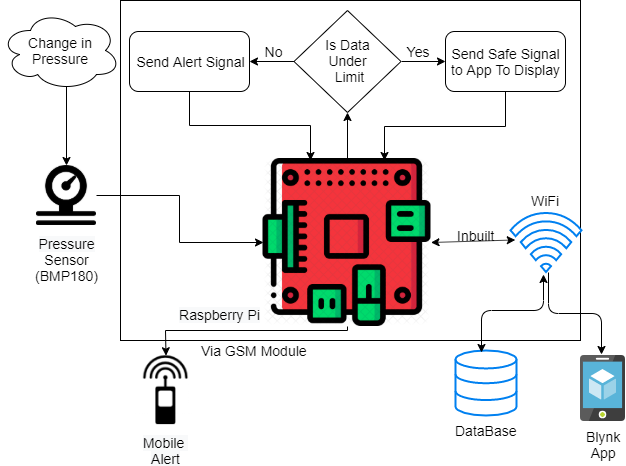
## Module 3



Description: Humidity Sensor(DHT11)

Humidity Sensor sends the data to Raspberry Pi and if the change in Humidity is not under the limit provided than it will send an Alert Signal to Mobile Phone via GSM Module and same to the BLYNK App via Wifi. If the change in Humidity is under the limit provided that it will send a Safe Signal to Mobile Phone via GSM and BLYNK App via Wifi.

## Module 4



Description : Pressure Sensor(BMP180)

Pressure sensor detects the pressure in the atmosphere and sends the data to Raspberry Pi and if the Data is under the limit which is defined then Alert Signal would be sent to Mobile Phone via GSM Module and Blynk App via Wifi. If the data provided by pressure signal is under the limit then it would send a safe signal to Mobile Phone via GSM Module and Blynk App via Wifi.

## Module 5

## 

Description : Temperature Sensor (DHT11)

Temperature Sensor detects the temperature of the atmosphere and sends the data to Raspberry Pi and if the Data is under the limit which is defined then Alert Signal would be sent to Mobile Phone via GSM Module and Blynk App via Wifi. If the data provided by pressure signal is under the limit then it would send a safe signal to Mobile Phone via GSM Module and Blynk App via Wifi.

**Data generation:**

Sensors Data:

1. BMP180 (Pressure sensor)

variables: sensor = BMP085.BMP085() // output value

Function: sensor. readpressure(), sensor.read\_sea\_level\_pressure()

2. DHT11 (Temperature and humidity)

variables and function : humidity, Temerature = Adafruit.DHT\_read\_retry(sensor,pin) // output value

3. Air\_quality sensor

variable GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN

function: GPIO.read() // output string

**Data Passing to Database:**

1. **Send Query:**

$temp = $\_GET["temp"];

$sql = "INSERT INTO `temprature` (`temp`) VALUES (";

$sql = $sql.$temp;

$sql = $sql.")";

echo $temp;

mysqli\_query($dbconnect,$sql);

1. **Receive Query:**

$records=mysqli\_query($dbconnect,$sql);

$json\_array=array();

while($row=mysqli\_fetch\_assoc($records))

{

$json\_array[]=$row;

}

1. **Connection with database:** $dbconnect = mysqli\_connect($server, $dbusername, $dbpassword);

Data is now stored and received from database using above queries.

1. **Connection with Blynk: sudo ./blynk-token=2432a5113a448fcb45e664a1934215**

This token is generated from blynk application on clicking new user

**Data Display**

On Blynk fetched by raspberry pi mapped to various sensor and personal configuration

# USE CASES

1. To set up: He/she has to download the BLYNK, he/she has to register, enter details and sync location. Give configurations such as thresholds to get mail/text alerts as per requirements.BLYNK app will display data.
2. To end/delete user details and his subscription: Log out of the app. let us know that they want to discontinue. We will delete all the user details from database permanently.
3. He/She when asks for the average data of 3 months, first of all it is checked whether he/she is a Subscribed user or not and if he/she is a subscribed user than they will be provided with the average data of 3 months and if not a subscribed user than it would not allow them to get the average data.
4. In a case where temperature goes up suddenly than if he/she has set a threshold after which the device will notify him/her through mobile message, we have ensured it because message was sent to him/her regarding the sudden increase in temperature.

# CONSTRAINTS

* If the display device on which the alert is to be generated is a mobile phone, Blynk App (Open Source Application - available on google playstore with the same name) must be installed.
* If the display device on which the alert is to be generated is a laptop/PC, Blynk Software Tool(Open Source Tool - available on the internet with the same name) must be installed.
* If the display device on which the alert is to be generated is using a web browser to display the alert, blynk.io (Website) must be open and the device linked with it.
* On start-up the user must connect the device with a display device using either of bluetooth or Wifi. If using bluetooth, the pairing with the device must be successful and if using wifi, the display device must connect to it’s Wifi.
* If the user is a premium user he/she must have a local host or server pre-installed for the storage of his/her own data values.

# ASSUMPTIONS

* User is aware about the concentration levels necessary for health safety when he/she changes the levels from default.
* All the sensors are functioning properly.
* Network connectivity is available constantly
* Power Supply is connected.
* Database is functioning properly and accessible only by user.
* The API call made to National value report receives correct values.

# APPENDIX

# Future Works

* In future a smaller embedded system can be made miniaturizing the system and show that a wearable device can also be made in future.
* Data Analyzing can be made more precise and can be analyzed according to your needs like we can get data of a specific region and it can be used to analyze lives of people residing in those regions.