DESIGN OF SMART SYSTEM AIDED BICYCLE

A PROJECT REPORT

Submitted by

AADITHYA V	**312213106001**
ADARSHANA RAJAGOPAL	**312213106003**
ASHWIN V	**312213106012**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SSN COLLEGE OF ENGINEERING, KALAVAKKAM 603 110

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2017

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"DESIGN OF SMART SYSTEM AIDED BICYCLE"** is the bonafide work of **"AADITHYA (312213106001), ADARSHANA RAJAGOPAL (312213106003), ASHWIN V (312213106012)"**, who carried out the project work under my supervision.

Dr. S. RADHA	N. PRABAGARANE
HEAD OF DEPARTMENT	**SUPERVISOR**
Department of ECE,	Assistant Professor,
SSN College of Engineering,	Department of ECE,
Kalavakkam – 603 110	SSN College of Engineering,
	Kalavakkam – 603 110

Submitted for examination conducted on ___________________________.

EXTERNAL EXAMINER INTERNAL EXAMINER

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project supervisor, **Mr. Prabagarane Nagaradjane**, for his unfailing support and enthusiasm throughout the duration of the project. We would like to thank him for imparting valuable knowledge and for constantly encouraging us towards contributing to the research community.

We would like to thank **Dr. S. Radha**, Professor and Head of the Department of Electronics and Communication Engineering, for her continuous support and appreciation.

We would also like to thank our project co-ordinators, **Dr. S. Sakthivel Murugan** and **Dr. R. Kishore**, who took a great deal of interest in our work and provided us with a lot of constructive criticism that helped us better the quality of our work. We take this opportunity to thank the staff of the department of ECE for their friendly assistance all throughout. Finally, we would like to thank family and friends for simply being there for us.

Aadithya V

Adarshana Rajagopal

Ashwin V

ABSTRACT

The Design of Smart System aided Bicycle consists of a standalone system which is a platform for the acquisition of real time data to aid the development of green field smart city and additionally the vitals of the user riding the bicycle. This project addresses pressing issues like air pollution monitoring, health monitoring and location tracking with the use of sensors that were seen as limitations in the already existing smart e-bike monitoring system (SEMS). Like SEMS, the above system is scalable, modular and replicable. This standalone system is used to monitor parameter such as location (GPS), air quality levels (ppm) and heart rate of the rider in real-time. It also addresses data security issue with the use of a fingerprint authentication and a GSM/GPS module to alert the user in case of system theft. The data thus collected is fed into Thingspeak, an open IoT platform powered by Matlab. The data is then visually represented in an android application for riders to view and analyse the same. The android application designed for the system assists the user with location tracking, pollution mapping and health monitoring. This project details the design and implementation of the hardware and software, enlists the outcomes of the system and explores the future scope and development. The system attached to the bicycle can be implemented on a large scale as a fleet/bike sharing system in order to map the entire city's pollution and employs Internet of Things (IoT) in the perspective of Intelligent Transportation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Connotations
AQI	Air Quality Index
API	Application Programming Interface
APK	Android Package Kit
BSS	Bicycle Sharing System
DAQ	Data acquisition
DDC	Display Data Channel
GPS	Global Positioning System
GSM	Global System for Mobile Communication
GPRS	General Packet Radio Service
GUI	Graphical User Interface
IDE	Integrated Development Environment
I2C	Inter Integrated Circuit
ITS	Intelligent Transport System
JSON	JavaScript Object Notation
MQTT	Message Queue Telemetry Transport
NMEA	National Marine Electronics Association
R&D	Research and Development
SEMS	Smart E-bike Monitoring System
SPI	Serial Peripheral Interface
TTFF	Time-To- First-Fix
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Intelligent transport system (ITS) has been one of the leading research areas in transportation which is aimed at improving the quality of services offered in different transportation modes. This is done with the help of various technologies such as wireless communication, detection techniques, sensing and computational technology.

ITS primarily aims at addressing collision avoidance, traffic management and driver assistance. The use of bicycles has been increasing and developmental technologies like Smart E-bicycles having been cropping up in European countries.

The European Government has stated that by 2020, the numbers of casualties on road will be halved because of increase in use of bicycles. The countries that have wisely invested in improving standards of bicycle infrastructure are expected to have the lowest casualties on road. Many countries like Canada, USA and almost all European countries have already made advancements by developing methods to resolve the issues in ITS. In this context, following are the countries that are very keen on ITS.

- Dubai has the fastest growing transportation in the world and has already been providing automatic traffic management plans to avoid congestion.
- Canada, one of the pioneers in ITS has seen controlled traffic systems on road as early as 1960s.
- The United Kingdom added Electronic toll collection and recycling old vehicles to reduce carbon footprint.

The resourcefulness of bicycles have been undermined and ignored for years now, until finally a few realize the compelling potential of the use of bicycles. The work done in Smart bicycles till date are motor powered driver assistance system called Pedlecs, to support cycling in hilly terrains and also to control the speed of vehicle, along with a mobile application for navigation [2]. The tremendous scopes for improvement are in areas such as addition of sensing modules, GPS tracking, environmental and health monitoring system [7].

1.1.1 Bicycle Sharing System

Bicycle sharing system (BSS) is a service that allows individuals to rent a bicycle out for a temporary period. This service is provided for free or a very feeble amount. The system can be traced back to Europe around 1950 when it first came into existence. They fall into two broad categories, one is the system promoted by a group of people called community bicycle programme and the other is promoted by government called the smart bicycle sharing programme. The bicycles are used to transit from one point to another.

1.2 MOTIVATION

Intelligent transport of systems has predominantly focused on cars. Smart bicycles are efforts inspired to make the bicycle more smart, unfortunately areas that lag behind capturing the attention of researchers is the use of sensing network to collect local environment details.

The motive of the Union Government of India is to develop Greenfield Smart and Sustainable cities within the next decade. The parameters we would like to address are pollution and environmental data, intelligent transportation system and eco-friendly sustainability of the bicycle sharing system [2].

The citizens can contribute to the building of smart city by collecting data through bicycle sharing system, important parameters such as Air Quality Index (AQI) along different areas and routes all over the city when they use it for the ride. Another feature to promote the use of such bicycles is through a customised app that collects data with which the user makes a self-analysis.

The rapidly rising population is consistently increasing the burden on roads causing congestion, poor air quality and other potential problems [1]. The development in ITS is subject to all the growing needs in the economy. Considering this, cycling is the most viable mode of transport with zero environment impact and affordable.

1.3. OBJECTIVE

- To build a standalone system consisting of sensors addressing security, pollution control and heart rate monitoring and integrate it to the bicycle.
- To develop a system that collects data, that is stored in Thingspeak database and is analysed to aid the cyclists.
- To construct an overall IoT system that consists of sensors and an Android powered mobile application that supports pollution mapping, air quality and health monitoring, tutorial videos and a chat room.
- To address a part of smart city module and offer a solution to collect the smart city parameters.
- To promote bicycle sharing system.

1.4 ORGANISATION OF THE THESIS

The thesis outline is as follows and divided into 5 chapters.

Chapter 1 gives Background, Motivation and Objective of the project and a brief introduction about ITS along with concept of Bicycle sharing systems is presented.

In chapter 2, we have the related works that put forward the various sources from which the project drew inspiration.

In chapter 3, we have the hardware design that aided us in improving the data collection and acquisition part of smart bicycles. It details the requirements and needs of the entire components used.

We begin chapter 4 with an overview of the hardware and detailed descriptions on the smart systems. The various components are reviewed and benefits of them are listed.

In chapter 5, we commence our discourse by disclosing the software used to create the application along with various activities such as YouTube, Chat Room and Navigation along with Firebase to create login and registration.

At the end, in chapter 6, we have the conclusions that discuss the areas covered in this project, results obtained and the sustainability of the system as the most viable and the future scope.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

The project drew inspiration from all the on-going research that exemplifies one of the most pressing issues of our time. The need for sustainable transportation system considering all the data is presented. The analysis includes stumbling upon data that reveal the drastic lagging in utilizing bicycles' resourcefulness. This is in comparison to China and few of the European countries.

Key reasons behind less cycle use lies in faulty parts, lack of premium raw materials, safety standards, cost of importing raw materials and lack of R&D. Also the government encourages cycling by providing dedicated cycling lanes and bicycle traffic signal heads that combines with conventional traffic to aid cyclists. The rest of the chapter details implementation of ideas on similar lines.

Smart E-bike Monitoring System (SEMS): Real time open source and open hardware GPS assistance and sensor data for electrically-assisted bicycles: E-bicycles or Pedlecs are mounted with sensors like GPS that is supported by an Android powered IOIO board and the measured real time data about weather from the users are integrated to a smart phone that can instantly share the data on to social media that helps other users [2]. Additionally, this system offers driver assistance that helps the user with pedalling in hilly terrains. It also limits the speed of the vehicle to 25 km/h.

This project captured the interest of cycling culture that exists in different countries across the globe. The design was tested in UK for duration of 10 months with about 30 cyclists. Certain areas of the project weren't addressed in SEMS, few of which were air quality monitoring, noise pollution monitoring and use of sensors to collect the same [2]. Another place of interest was monitoring the vitals of the user and torque sensor data. One of the pressing issues of the design was powering.

SEMS could be of interest to a range of stakeholders, including e-bicycle manufacturers, battery manufacturers and those purchasing or using fleets. Fleet management of public or private e-bicycle fleets could also be part of future developments. The scope of development is immense, especially considering it for bicycle sharing systems.

A Security-aware Safety Management Framework for IoT-integrated Bicycles: The idea is to integrate sensors to avoid mishaps whilst the vehicle is in use. For instance, a humidity sensor reads the humidity in the car that may lead to drowsiness. All the parameters are recorded for references by the company. Taking another instance, a drunk driver's behaviour reflects in his mobility. The availability of data for self-analysis of driving on a smart phone app is a boon for any driver. Live alerts could be presented to the user while driving in order to ensure safety by sensing the environment [6]. Data collection and analysis could be done for different scenarios and messages can be accordingly delivered on the Android application. By integrating different values and different situations, a driving pattern for a user can be presented. Parameters such as fuel efficiency, CO_2 emission etc., could be measured to make it a more enhanced application, letting the user know the nuances of his/her driving [9].

A Mobile GPRS enabled Air quality monitoring: This paper offers a solution to measurement of air quality with the help of General Packet Radio Services (GPRS) enabled sensor arrays. A Mobile Data Acquisition (DAQ) system is built consisting of a single-chip microcontroller, air pollution sensors, GPRS-Modem and a Global Positioning System (GPS) Module [1]. A pollution server is maintained where the data is collected is heaped. The gases gathered are (CO, NO_2, and SO_2) and the mobile DAQ sends it to the server along with GPS coordinates and time. The data can be accessed by interest groups such as environment protection agencies, vehicles registration authorities, and tourist and insurance companies. The data can be integrated to Google Maps to display the pollution values in real time. The gases measured in Parts Per Million are converted to AQI for maintaining standards. The same idea is drawn here along with the implementation with regards to SEMS [2]. The pollution mapping is analogous to the working of Google maps. Anonymous data is fed back from the user to the cloud that determines the presence of all such users to provide information about traffic. Similarly, if more people come forward to embrace bicycle sharing system that uses the smart bicycle dedicated to collecting such environment variables, the accuracy of the system will be scaled up [8].

2.2 SUMMARY

This chapter presented an overview of the related works in the field of smart system aided bicycles. It discussed in detail, the areas that need to be addressed leading to the idea of this project.

CHAPTER 3

HARDWARE REQUIREMENTS

3.1 INTRODUCTION

The main concern for the system design was scalability, replicability and modularity. The purpose of developing such a system is to promote the bicycle sharing systems in the future. The system, as seen in Figure 3.1 was designed and implemented using open source hardware and software to permit further development of such a system. Other requirements for designing the system are data and engineering.

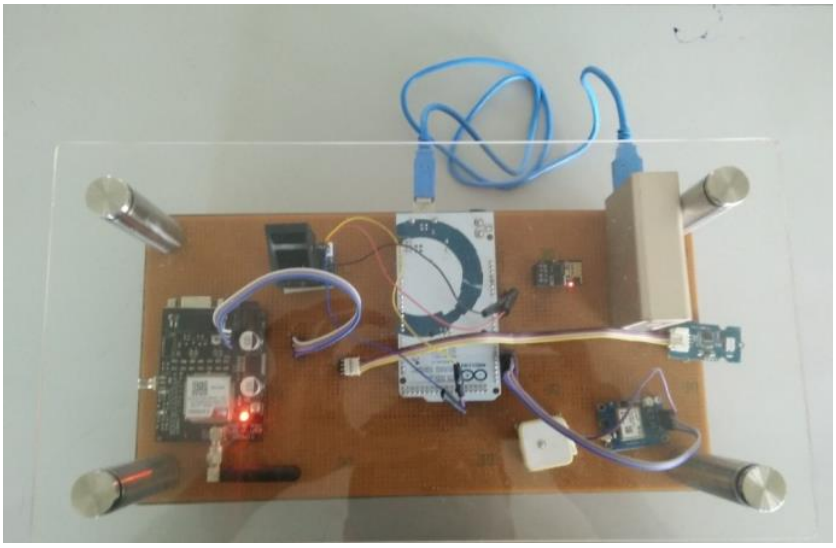


Figure 3.1 Complete hardware integrated in dotted board

3.2 DATA REQUIREMENTS

The system requires tracking of the cycle’s location (latitude, longitude, time, data and velocity) so as to document the trip in detail and to keep a check on the whereabouts of the cycle. Another major data that needs to be collected and analysed is air quality levels. An extensive database about polluting gases is required for the planning of Greenfield smart cities in order to check the menace of dropping air quality level. Cycling is one of the best forms of exercise and monitoring heart rate is essential to people who exercise in order to measure the intensity of the workout. Maintaining a consistency is essential to make the best out of cycling.

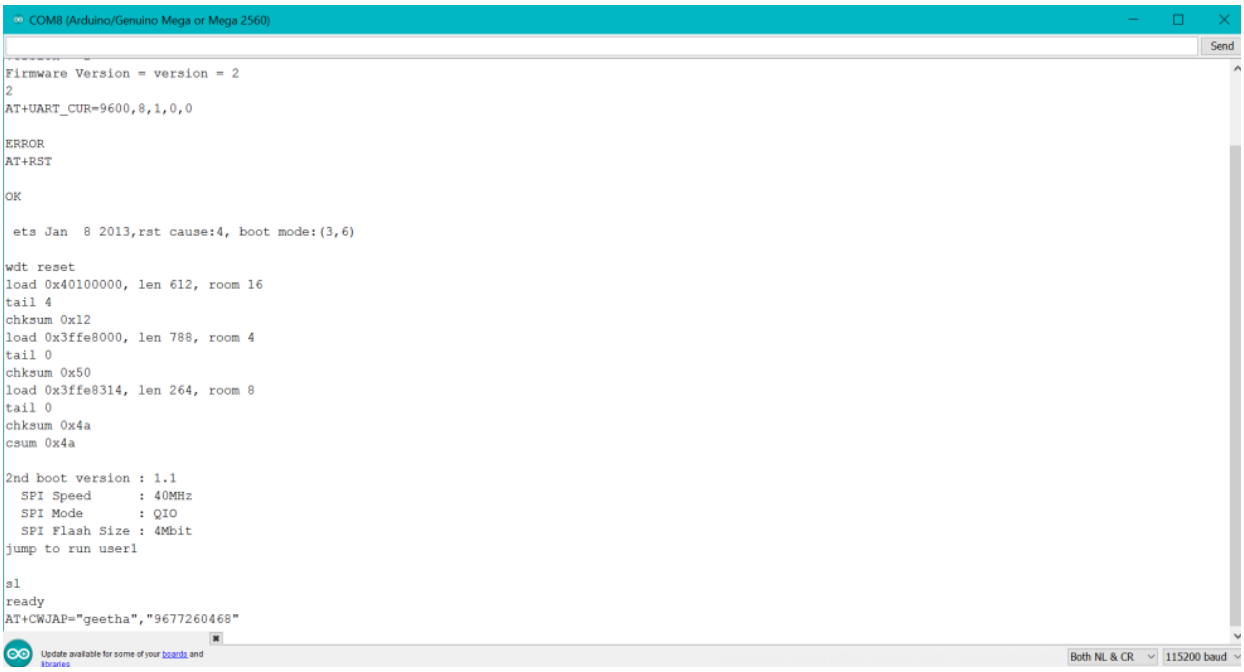


Figure 3.2 Serial monitor dataset 1

Figure 3.3 Serial monitor data set 2

The system needs to collect real time information as shown above in Figure 3.2 and Figure 3.3 which will aid the user by displaying it in Android application.

The real time pollution data can be mapped on to the route in Google Maps once the ride is completed. The data can also be parsed to give a graphical representation. As a fleet undertaking different routes, the entire city's pollution levels can be monitored.

3.3 HARDWARE DESIGN

The smart cycle is broadly split into two components: hardware and software. The first section is the hardware comprising of microcontroller, sensing elements and data acquisition modules. The basic purpose of the hardware is to collect real time data and send it to the cloud for further analysis. The key feature of the system is the security element that is enabled by a fingerprint sensor and is backed up by GSM/GPS module that alerts the user in case of an unfortunate situation. The entire hardware design is shown in Figure 3.4.

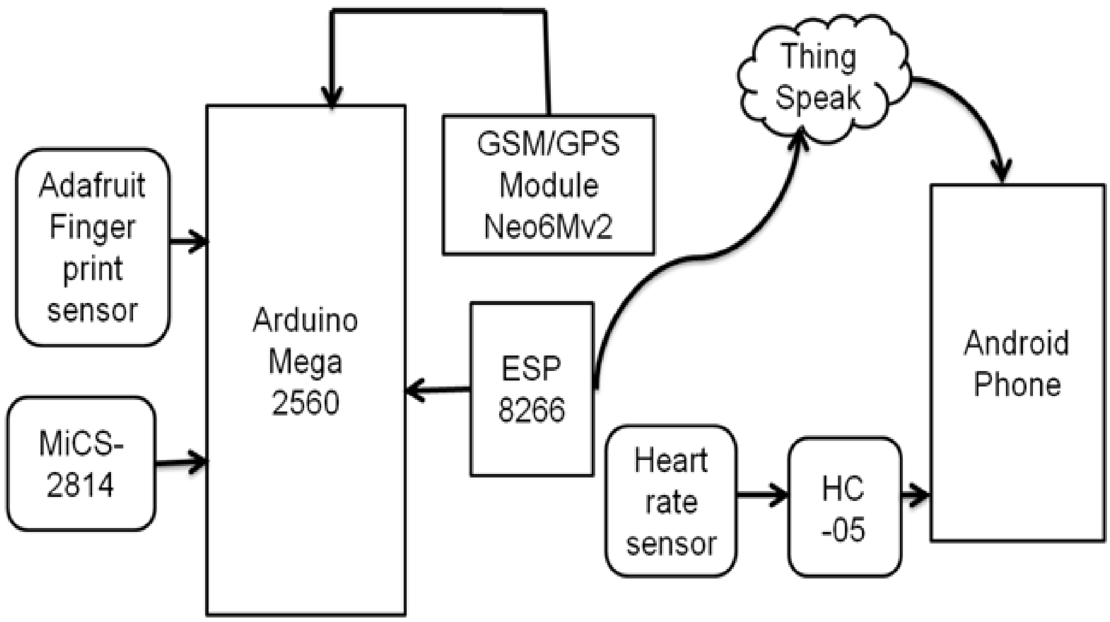


Figure 3.4 Block diagram of Hardware

3.3.1 Powering the System

The system has a very specific working voltage range. It is necessary to supply constant power supply and current to all the components present. The Mega 2560 has to operate on an external supply of 6V-20V and is then regulated to 5V. If the board is supplied with less than 7V, less than 5V is regulated to the board and it destabilises the board and malfunctions. On the other hand when supplied with greater than 12V, the voltage regulator overheats and damages the board. The Arduino Mega is powered by using a 5000 mAh power bank. It is necessary to power the Arduino using a power bank because the various peripherals are powered using the power pins on Arduino. The various peripherals and their power requirements are listed in the Table 3.1.

Table 3.1 Components and their power requirements

Peripherals	Operating Voltage (volts)	Operating Current(mA)
Fingerprint Sensor	5	120
GSM	5	500
GPS	5	45
ESP-8266	3.3	60 uA-0.5mA
Grove Multichannel Gas Sensor	5	17.5Ma
Pulse Sensor	5	4mA
HC-05	5	35mA

- GND: The reference point from which the voltage difference is measured
- 5V: It is used to power sensors. This is generated by an on-board regulator.

- 3.3V: A 3.3 volt supply generated by the on-board regulator. Maximum current drawn is 50 mA.

3.3.2 Components

The smart system is based around an android phone along with Arduino mega board. The Mega 2560 is an upgraded version of the Arduino Uno R3 with higher number of digital I/O pins, Analog inputs and a higher memory for higher programming capabilities. It features 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The ATmega2560 possesses 256KB of flash memory for storing and processing the code, 8 KB of SRAM and 4KB of EEPROM. 8KB of the flash memory is used for the boot-loader.

3.4 DEPLOYMENT OF HARDWARE

The entire hardware is integrated together in a dotted board which is placed in a wooden enclosure covered by a transparent acrylic sheet. The system is attached to the carrier of the bicycle as shown in Figure 3.5.The components used are briefed along with their specification.

Figure 3.5 Smart System Aided Bicycle

3.4.1 GPS Neo-6m u-blox

The NEO-6m u-blox GPS module is a standalone GPS receiver embedded with the high performance u-blox 6 positioning engine. The 50-channel u-blox 6 positioning engine consists of a Time-To-First-Fix (TTFF) of less than 1 second. The dedicated acquisition engine, with 2 million correlates, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Neo-6m GPS receivers have excellent navigation performance even in the most challenging environments. The Neo 6m possesses a standalone GPS runs on 2.7V-3.6V, supports UART, USB, SPI, DDC, uses a crystal oscillator, possesses a RTC crystal, 3 configuration pins, 1 time pulse and external interrupt, external Antenna supply and supervisor. The block diagram of the GPS module is as shown below in Figure 3.6.

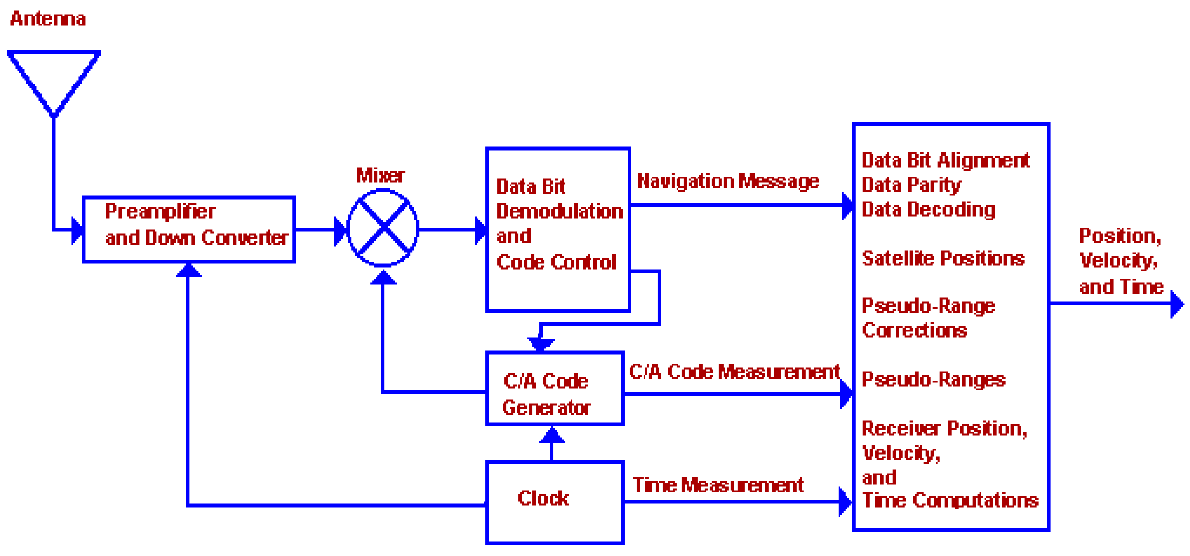


Figure 3.6 GPS Neo-6M Block diagram

3.4.1.1 Protocol

The Neo ublox-6m uses the NMEA 0183 protocol which is a combined electrical and data specification for communication between marine electronic devices. It has been defined and controlled by the National Marine Electronics

Association. The NMEA standard uses a simple ASCII, serial communications protocol that defines the way in which data is transmitted from one 'talker' to multiple listeners [5]. Using a multiplexer, a computer port can communicate with multiple sensors. At the application layer, the standard also defines the contents of each sentence (message) type, so that all listeners can parse messages accurately.

3.4.2 GSM SIM 808

SIM 808 is a Quad-Band GSM/GPRS module which combines GPS technology for satellite technology. It features an industry-standard interface and GPS function; it allows variable parameters to be measured at any location with signal coverage. SIM808 can be used for phone calls, SMS, connecting to the internet and finding GPS co-ordinates. Our functioning of the GSM SIM 808 is to send an emergency SMS. The entire block diagram of the GSM SIM 808 is described in Figure 3.7

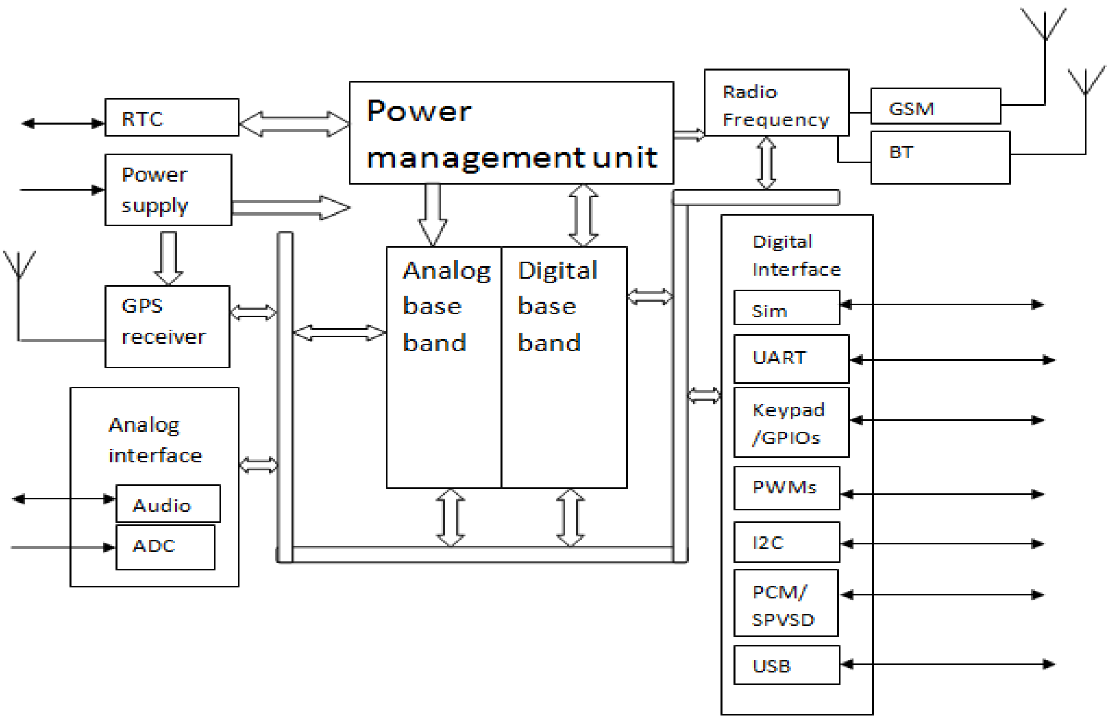


Figure 3.7 GSM SIM808

The SIM card mounted on GSM modem, upon receiving digit command by SMS from any cell phone sends that data to the MC through serial communication. While the program is executed, the GSM modem receives command 'STOP' to develop an output at the MC, the contact point of which are used to disable the ignition switch. The command sent by the user is based on an intimation received by him through the GSM modem 'ALERT', a programmed message only if the input is driven low.

3.4.3 Grove Multi channel Gas Sensor

Grove – Multichannel Gas sensor is an environment detecting sensor with a built in MiCS-6814 which can detect many unhealthy gases, and three gases can be measured simultaneously due to its multi channels, so it can help to monitor the concentration of more than one gas.

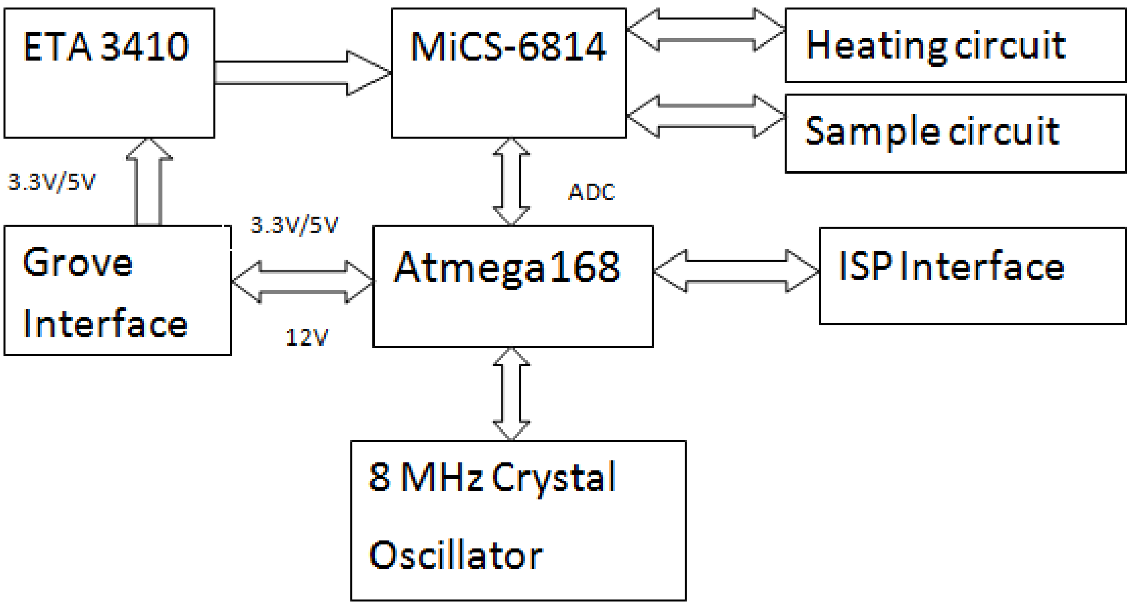


Figure 3.8 MiCS 6814

3.4.3.1 Features

The various features as shown in the Figure 3.8 include

- Three fully independent sensing elements on one package
- Built with ATmega168PA
- I2C interface with programmable address
- Heating power can be shut down for low power

Detectable gases:

- Carbon monoxide CO 1 – 1000ppm
- Nitrogen dioxide NO_2 0.05 – 10ppm
- Ethanol C_2H_6OH 10 – 500ppm
- Hydrogen H_2 1 – 1000ppm
- Ammonia NH_3 1 – 500ppm
- Methane CH_4 >1000ppm
- Propane C_3H_8 >1000ppm

3.4.4 ESP

Espressif Systems' Smart Connectivity Platform is a set of high performance, high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

ESP8266 EX-01 has been designed for mobile, wearable electronics and IoT applications with the aim of achieving the lowest power consumption with a combination of several proprietary techniques. The power saving architecture operates mainly in 3 modes: active mode, sleep mode and deep sleep mode. By using advance power management techniques, logic to power-down functions are not required. ESP8266 EX consumes about than 60uA in deep sleep mode (with RTC clock still running) and less than 1.0mA (DTIM=3) or less than 0.5mA (DTIM=10) to stay connected to the access point. When in sleep mode, only the calibrated real-time clock and watchdog remains active. The real-time clock can be programmed to wake up the ESP8266 EX at any required interval. The block diagram of ESP is shown in Figure 3.9.

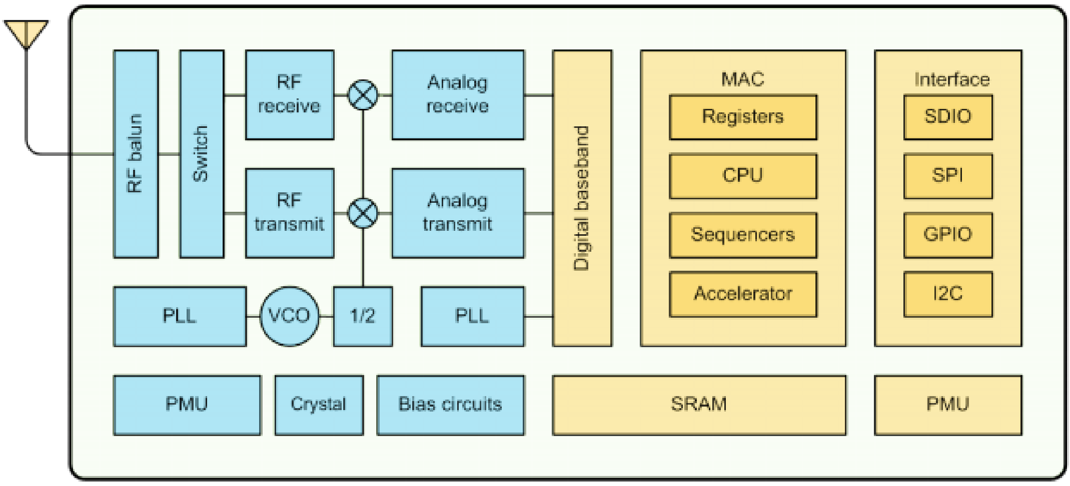


Figure 3.9 ESP Block diagram

3.4.4.1 Firmware

The application and firmware is executed in on-chip ROM and SRAM, which loads the instructions during wake-up, through the SDIO interface, from the external flash. The firmware implements TCP/IP, the full 802.11 b/g/n/e/i WLAN MAC protocol and Wi-Fi Direct specification. It supports not only basic service set operations under the distributed control function but also P2P group operation compliant with the latest Wi-Fi P2P protocol. Low level protocol functions are handled automatically by ESP8266:

• RTS/CTS

• Acknowledgement

• Fragmentation and defragmentation

• Aggregation

• Frame encapsulation (802.11h/RFC 1042)

• Automatic beacon monitoring / scanning

• P2P Wi-Fi direct

3.4.5 Adafruit Fingerprint Sensor

The Adafruit Finger print sensor is used for fingerprint verification and authentication. Usually used as biometrics for identification purpose, the same purpose is extended here. The Work flow is explained in Figure 3.10. The sensor acts as the gateway to the stand alone system. This marks the beginning of the process and a wrong authorization triggers the GSM to send a message to the mobile.

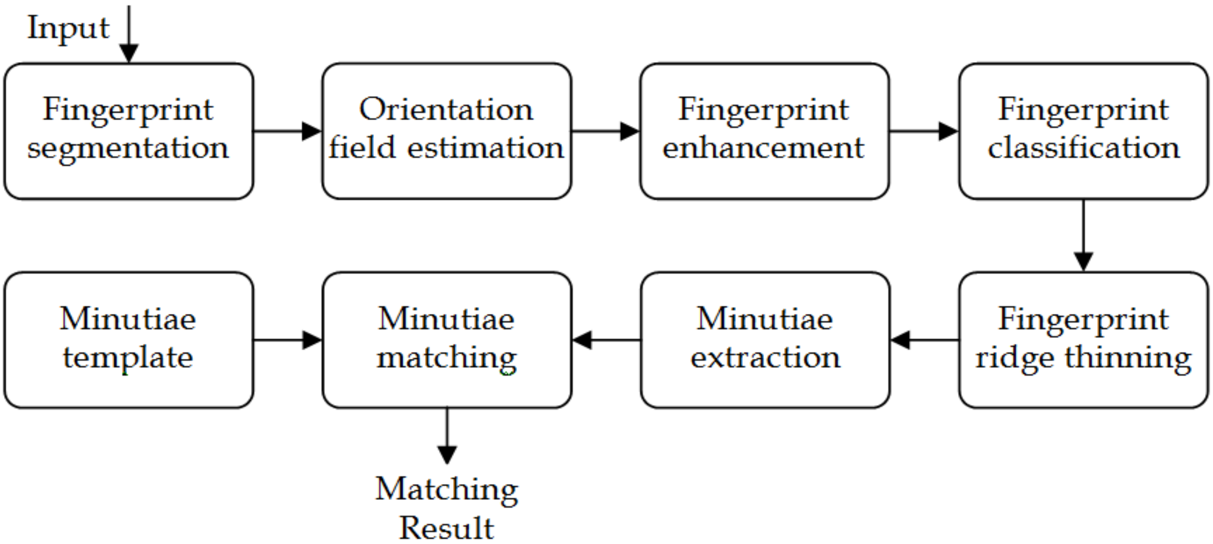


Figure 3.10 Adafruit Fingerprint Sensor workflow

3.4.6 Pulse Rate Sensor and Bluetooth

Pulse Sensor Amped is a plug and play heart-rate sensor for Arduino. It is an easy heart rate calculator which can be used by anyone, normal people and athletes, mobile developers. Pulse Sensor includes noise cancellation and amplification circuitry for providing exact readings. It is integrated with the

Bluetooth sensor to send it to the application. HC-05 Bluetooth Module is an easy to use Bluetooth SPP (Serial Port Protocol) Module, designed for transparent wireless serial connection setup.

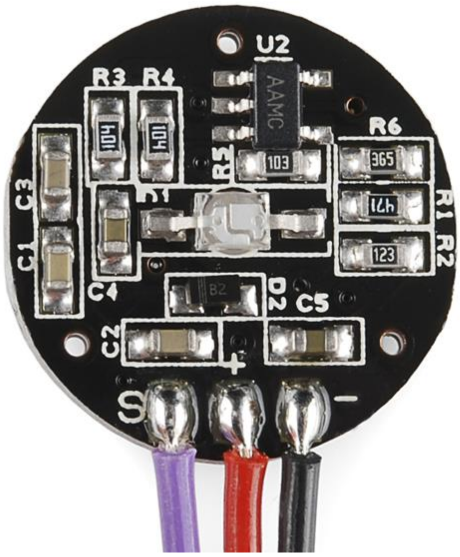


Figure 3.11 Heart Rate Pulse Sensor

Figure 3.12 HC-05

Its communication is via serial communication which makes an easy way to interface with controller or PC. HC-05 Bluetooth module provides switching mode between master and slave mode.

The sensor consists of a green bright LED and a light detector. When the heart pumps a pulse of blood through the blood vessels, the finger becomes opaque and so light is blocked. With each heart pulse, the detector identifies the pulse beat and calculates the heart rate for some period of time and estimates the output in BPM. The noise cancellation and the amplification circuitry as shown in Figure 3.11 come into play and increase the output signal.

3.5 SUMMARY

The algorithm for the entire system has been presented above. There are two separate parts to the algorithm of the system. The components described above are integrated in the dotted board and is connected to the I/O peripherals in the Arduino. On powering the system, the GPS waits for the signal from the satellite and the fingerprint sensor blocks ESP and air quality sensor from collecting the data. On placing the right authentication, the sensors start collecting the data and ESP stores it in the Thingspeak cloud. If there is a system security breach, GSM sends the location details to the registered mobile number. The software portion of the algorithm involving has been explained in detail in Chapter 4.

CHAPTER 4

SOFTWARE DESIGN

4.1 INTRODUCTION

The software forms the other major part of the smart bicycle. There are four major layers to the software design

- Arduino IDE
- Thingspeak
- Android Studio
- Firebase

Arduino IDE is used to collect the data from the sensors and write it to the cloud. Thingspeak acts as a bridge between hardware and the android application. Android Studio is the tool used here to design, develop and test the android application. The Android Application provides the visual representation of all the values stored on the cloud and improves the appeal to the user. With a range of features, the Android application enables the user to understand and analyse the environment parameters and heart monitoring. The data in the android application is secured through the use of Firebase login system. Unregistered users can register first through E-mail and password after which the various options can be chosen by tapping on the navigation drawer.

4.2 ARDUINO IDE

Arduino IDE is open source software for editing, compiling and burning the code into the Arduino Code. The Integrated Development Environment (IDE) contains a text editor, a message area, a text console, a toolbar with buttons for all purposes and menus. It is used to communicate with all the Arduino/Genuino boards to upload programs. It runs on Windows, Mac OSX and Linux. It is completely written in Java.

4.2.1 Process of Uploading

The Java sketches are written in the text editor and are saved with .ino extension. There are options to compile the sketches and then upload them to the board. The message area gives feedback and the errors. The console displays messages about compilation errors, uploading errors and other information. There are options to view the serial monitor and choose the COM port.

4.3 THINGSPEAK

Thingspeak is a Matlab powered IoT platform that performs data analytics. There are three major parts to it.

- Data collection
- Data analysis
- Action based on data analytics

4.3.1 Features

Thingspeak Features

- Data collection
- Open API
- Alerts
- Event Scheduling
- Matlab analytics and visualization
- App integrations
- Worldwide Community

Thingspeak works with

- Arduino
- Particle photon and electron
- Raspberry Pi

- Electric Imp
- Mobile and web Apps
- Twitter
- Twilio

4.3.2 Data Collection

A channel is created after an account is created in Thingspeak. Various fields can be mapped on to the same channel. Every channel has a unique Channel ID. The views and display options can be altered by the use of channel settings. Once the channel is created the data can be sent and retrieved and can be made it public to share data.

The open Application Programming Interface (API) is a set of definitions, rules and protocols for building a software tool. APIs are also used while programming Graphical user interface (GUI). For instance, using the REST API calls such as GET, POST, PUT and DELETE, we create a channel, update it on timely basis, clear an existing channel and delete a channel.

A typical channel has the following settings:

- REST API
- Use REST API calls to create and update Thingspeak channels and charts
- MQTT API
- Use the MQTT API to update Thingspeak channels

4.3.3 Data Analysis

Matlab supports algorithm that can integrated to the data collected from the channel to that enables you to explore, view and manipulate the data. It also helps us connect to social media such as Facebook, twitter and other media services. We use Thingspeak as an IoT Cloud storage mechanism in order to store values in the channels and then obtain them as a Java Script Object Notation (JSON) file to be later parsed in the android application and then drawn as a graph in the android application.

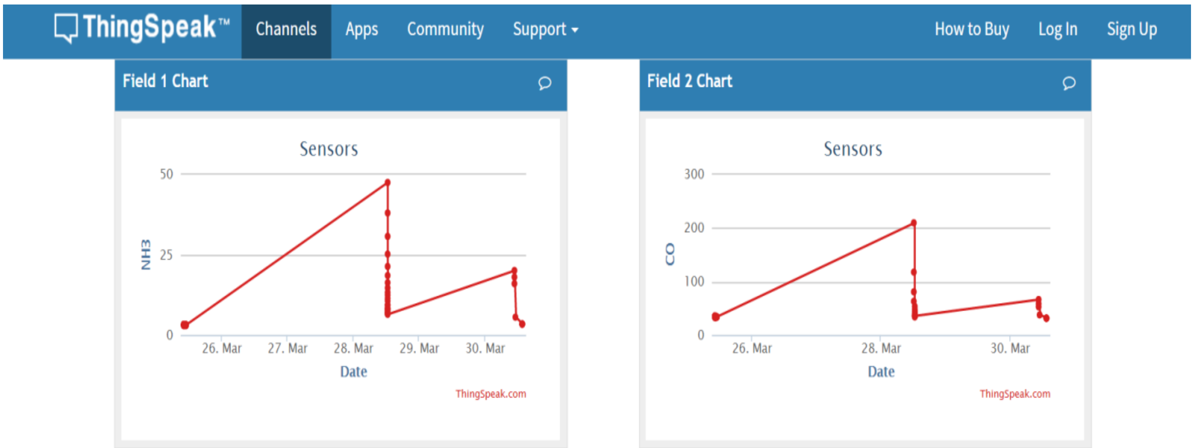


Figure 4.1 Thingspeak channel feed data set 1

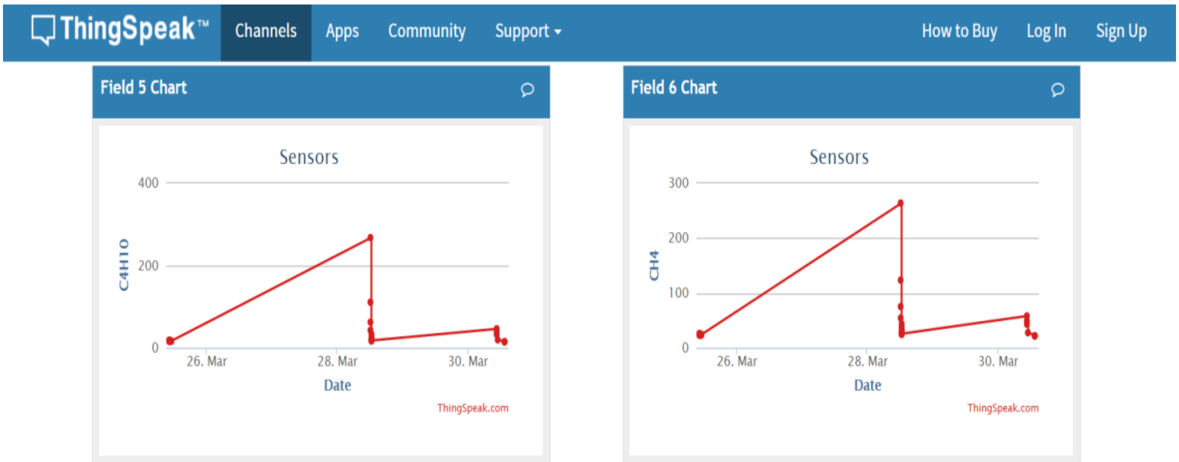


Figure 4.2 Thingspeak channel feed data set 2

Figures 4.1 and 4.2 are the channel feeds from the channel. The first six channels are dedicated for storing the gas values, whereas the last two channel feeds store the latitude and longitude. The channels are updated every 15 seconds. The X axis is always the date and time of the value. The Y-axis is dynamic with every value range. The latitude and longitude are in degrees, whereas the gas concentrations are in parts per million (ppm). The values keep updating as long as ESP 8266 is connected to the internet.

Secure | https://thingspeak.com/channels/223182/field/3.json

{"channel":
{"id":223182,"name":"Sensors","latitude":"13.8","longitude":"80.75","field1":"NH3","field2":
"CO","field3":"NO2","field4":"C3H8","field5":"C4H10","field6":"CH4","field7":"LATITUDE","fie
ld8":"LONGITUDE","created_at":"2017-02-05T04:23:53Z","updated_at":"2017-03-
30T08:12:43Z","last_entry_id":349},"feeds":[{"created_at":"2017-03-
25T04:35:16Z","entry_id":250,"field3":"0.181816"},{"created_at":"2017-03-
25T04:35:39Z","entry_id":251,"field3":"0.181816"},{"created_at":"2017-03-
25T04:36:02Z","entry_id":252,"field3":"0.183021"},{"created_at":"2017-03-
25T04:36:25Z","entry_id":253,"field3":"0.184228"},{"created_at":"2017-03-
25T04:36:48Z","entry_id":254,"field3":"0.183021"},{"created_at":"2017-03-
25T04:37:11Z","entry_id":255,"field3":"0.184228"},{"created_at":"2017-03-
25T04:37:34Z","entry_id":256,"field3":"0.185439"},{"created_at":"2017-03-
25T04:37:57Z","entry_id":257,"field3":"0.185439"},{"created_at":"2017-03-
25T04:38:20Z","entry_id":258,"field3":"0.183021"},{"created_at":"2017-03-
25T04:38:43Z","entry_id":259,"field3":"0.177027"},{"created_at":"2017-03-
25T04:39:06Z","entry_id":260,"field3":"0.177027"},{"created_at":"2017-03-
25T04:39:29Z","entry_id":261,"field3":"0.171106"},{"created_at":"2017-03-
25T04:39:52Z","entry_id":262,"field3":"0.169930"},{"created_at":"2017-03-
25T04:40:15Z","entry_id":263,"field3":"0.168757"},{"created_at":"2017-03-
25T04:40:38Z","entry_id":264,"field3":"0.167587"},{"created_at":"2017-03-
25T04:41:01Z","entry_id":265,"field3":"0.166420"},{"created_at":"2017-03-

Figure 4.3 JSON data format

All the data from the channel feeds are available in three data formats. JSON, XML and CSS. We use the JSON file as seen above in Figure 4.3 for parsing the graph onto the application for graph activity.

4.4 ANDROID STUDIO

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Instant Run to push changes to your running app without building a new APK
- Code templates and GitHub integration to build common app features and import sample code
- Lint tools to catch performance, usability, version compatibility, and other problems

- Each project in Android Studio contains one or more modules with source code files and resource files.

4.4.1 Gradle Build System

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. Some of the features of the build system are:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app, with different features using the same project and modules.

Android Studio build files are named build.gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When an existing project is imported, Android Studio automatically generates the necessary build files. The gradle file of this project is shown below in Figure 4.4

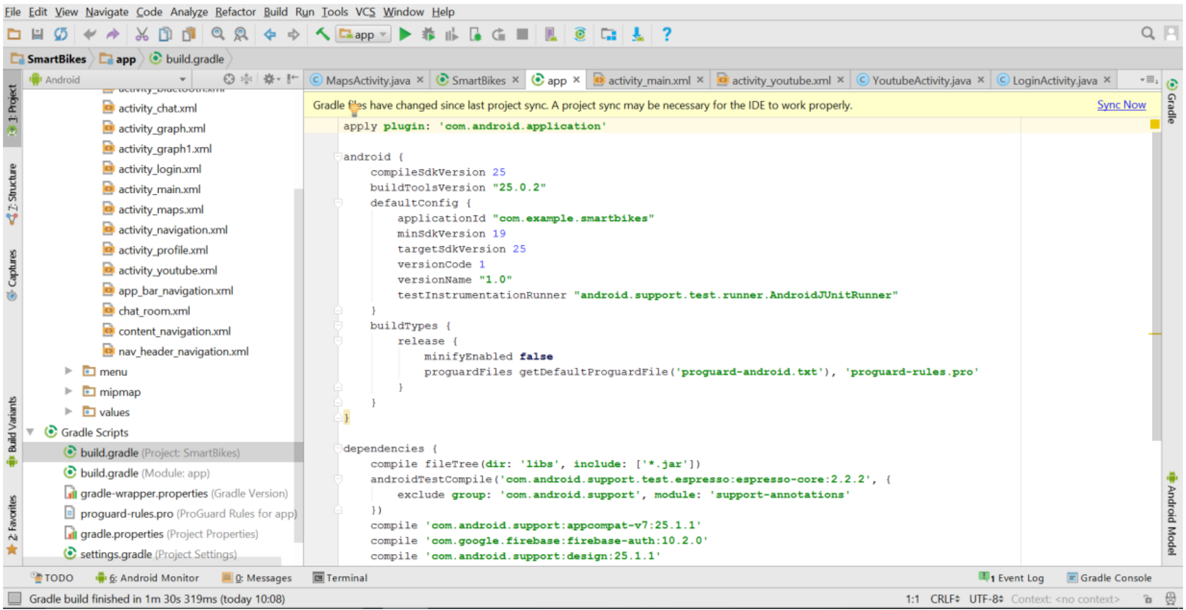


Figure 4.4 Build gradle (app level module)

4.4.2 Managing Dependencies

Gradle takes care of finding dependencies and making them available in the build. Module dependencies, remote binary dependencies, and local binary dependencies can be declared in build.gradle file. Android Studio configures projects to use the Maven Central Repository by default. This configuration is included in the top-level build file for the project. For any activity to perform specific functions, it is necessary to add the particular file to the dependency function in app level build gradle file. The dependencies required for the execution of this project is listed below in Figure 4.5.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.1.1'
    compile 'com.google.firebase:firebase-auth:10.2.0'
    compile 'com.android.support:design:25.1.1'
    compile 'com.google.android.gms:play-services:10.2.0'
    compile files('libs/GraphView-4.2.1.jar')
    testCompile 'junit:junit:4.12'
    compile 'com.google.firebase:firebase-database:9.0.2'
    compile 'com.google.firebase:firebase-core:9.0.2'
}
apply plugin: 'com.google.gms.google-services'
```

Figure 4.5 Dependencies, libraries and plugins

4.4.3 Activities

4.4.3.1 Map Activity

This Activity helps the user to show various routes and also map real time pollution over the route. It also allows the user to draw markers for starting and destination point. The pollution mapping can be done as shown below in Figure 4.6. The user first needs to download the JSON data in the graph activity before the mapping can be done. Inside the maps activity, the users have to choose the starting and destination point and the route will be drawn indicating the pollution levels. Red indicates highly polluted areas and Green indicates least polluted areas.

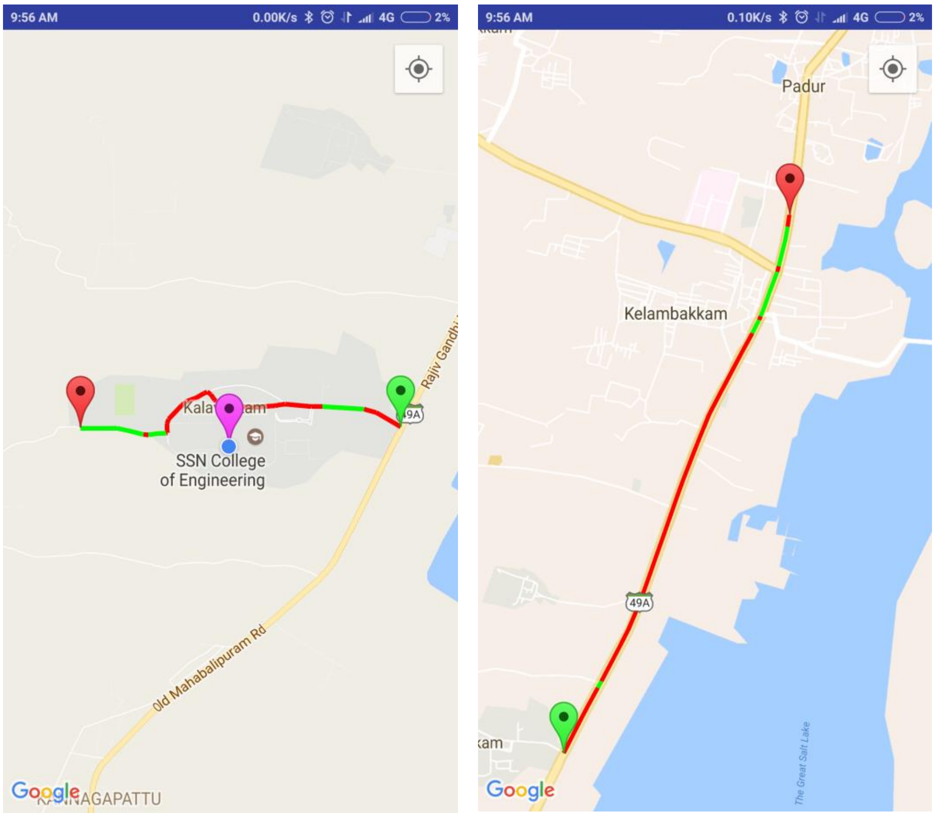


Figure 4.6 Pollution Mapping

4.4.3.2 YouTube Activity

The purpose of having a YouTube Activity in the Android Application is to enable the users to learn cycling to better themselves at the art of riding the cycle. This activity consists of a layout with specific links to Professional Cycling Channels. The layout lists down the several videos offered to the user by linking to the YouTube App in the Android Phone. YouTube's offline features make it easier to save data charges. The Java part consists of making the texts as active URL's to the specified tutorials as shown in Figure 4.7.

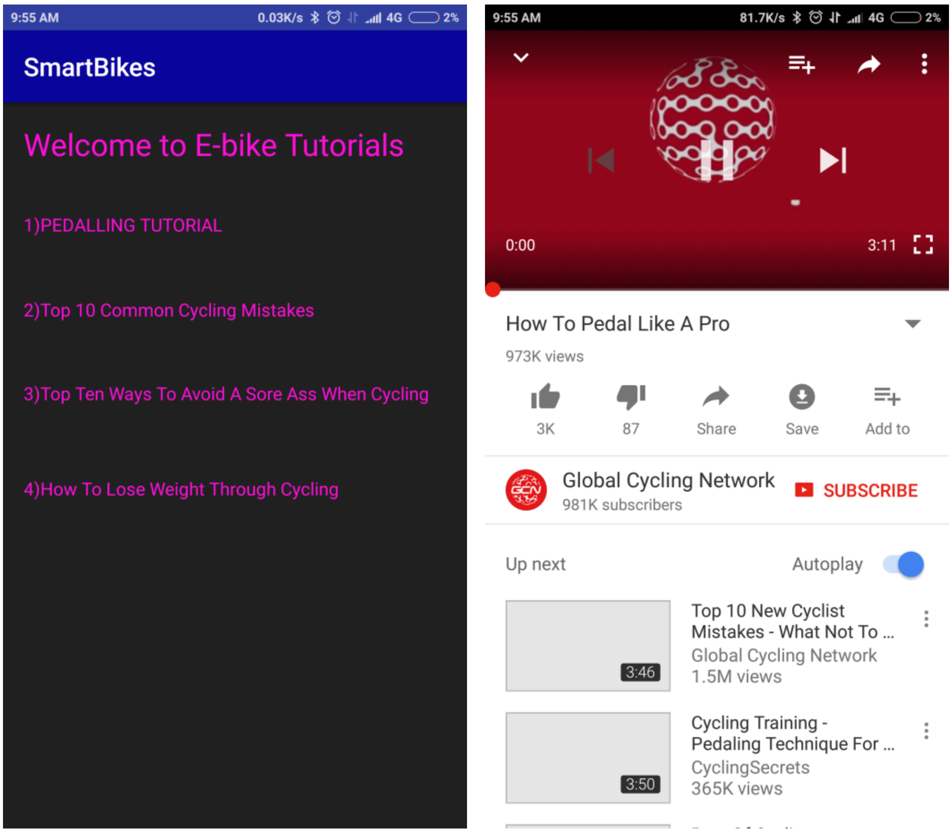
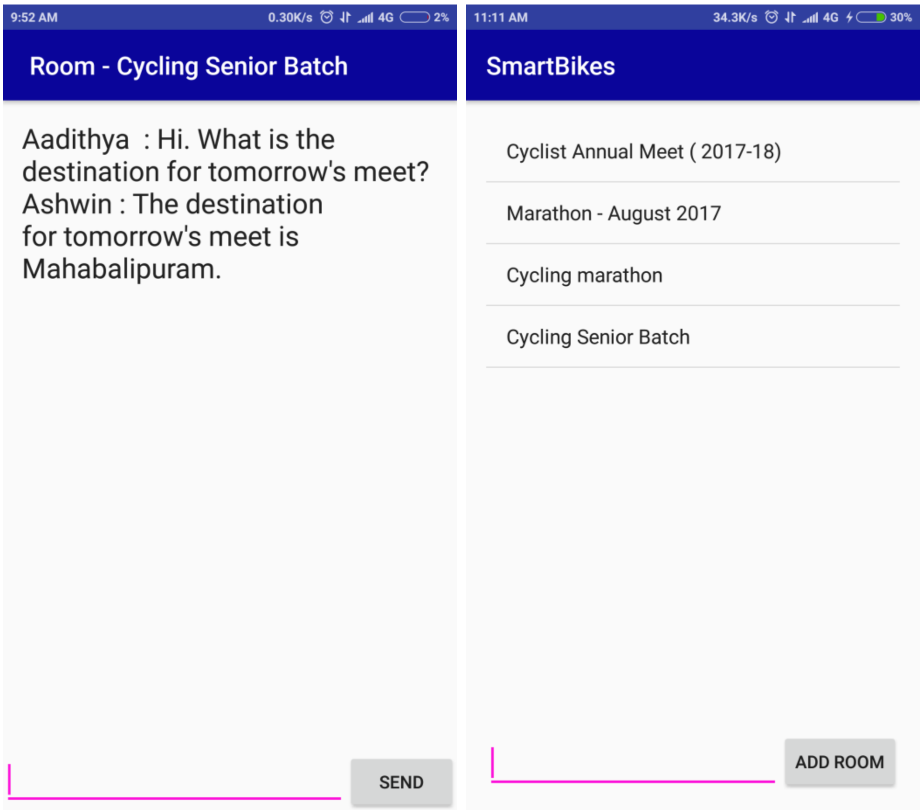


Figure 4.7 YouTube cycling tutorials

4.4.3.3 Chat Activity

This activity allows the user to communicate with other cyclists. Important information like Change of route, Marathon Plan and tracking other cyclists can be shared in this activity. The chat activity is created with the help of firebase database system. It allows the users to create special chat room/groups where people can discuss about all their day to day cycling plans. The users also have to enter their chat name. Chat activity makes use of firebase login system and the messages are secured with the same system. The chatting facility requires constant internet connection. The Chat Activity is shown in Figure 4.8 (a) and (b).



(a) (b)

Figure 4.8 (a) Chat room (b) Groups available in chat activity

4.4.3.4 Bluetooth Activity

The Bluetooth Activity is present to collect the heart-rate from Pulse-Sensor connected to Arduino via HC-05. There are two main parts to the Activity:

- Establishing Bluetooth connection
- Reading the values from HC05

The permission for turning on Bluetooth has to be added in the manifest file. On tapping the Heart Rate drawer, the Bluetooth is turned on in the Android mobile phone. The HC-05 on being powered specifies the Bluetooth address. The heart rate sensor connected to the Bluetooth HC-05 then transmits the values to the application and is displayed on the screen continuously as shown in Figure 4.9.

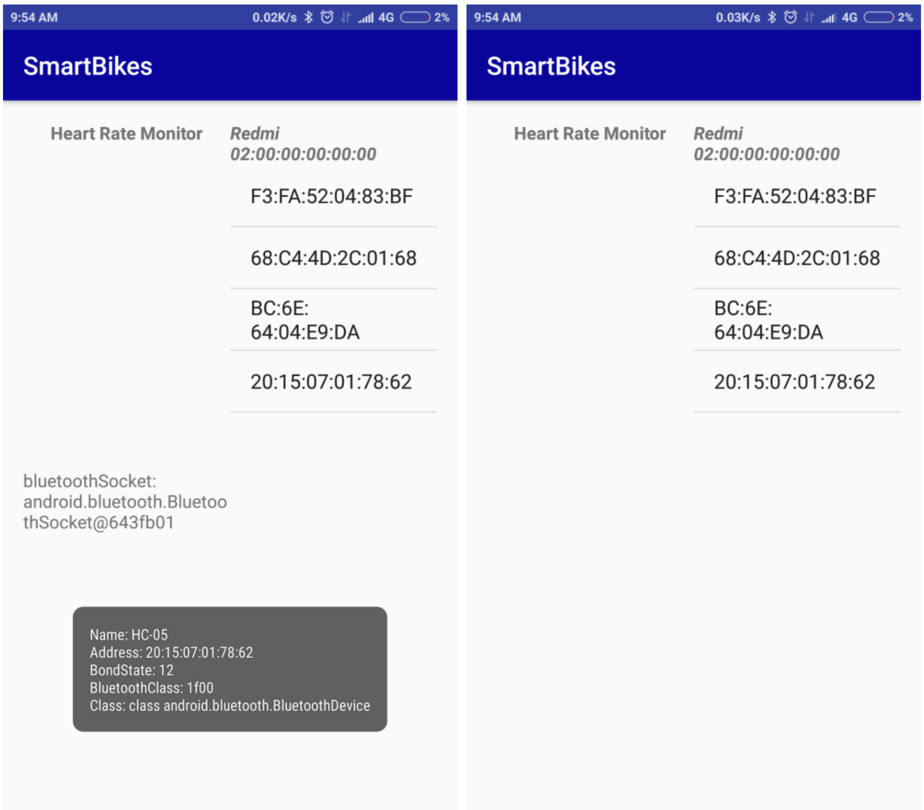


Figure 4.9 Bluetooth heart rate monitor

4.4.3.5 Graph Activity

This activity downloads the JSON data from Thingspeak and plots the pollution data having the data entry as x-axis and the pollution in ppm as y-axis. Each vehicle pollutant gas has different AQI and with the known breakpoints, AQI is calculated with the below mentioned formula for each ride and it is reported to the user in the same activity.

$$I_P = \frac{I_{Hi} - I_{Lo}}{BP_{Hi} - BP_{Lo}}(C_P - BP_{Lo}) + I_{Lo}$$

Where I_p is the index for pollution p

C_p-The rounded concentration of pollutant p

BP_{Hi} -The breakpoint that is greater than or equal to C_p

BP_{Lo} -The breakpoint that less than or equal to C_p

I_{Hi}-The AQI value corresponding to BP_{Hi}

I_{Lo} -The AQI value corresponding to BP_{Lo}

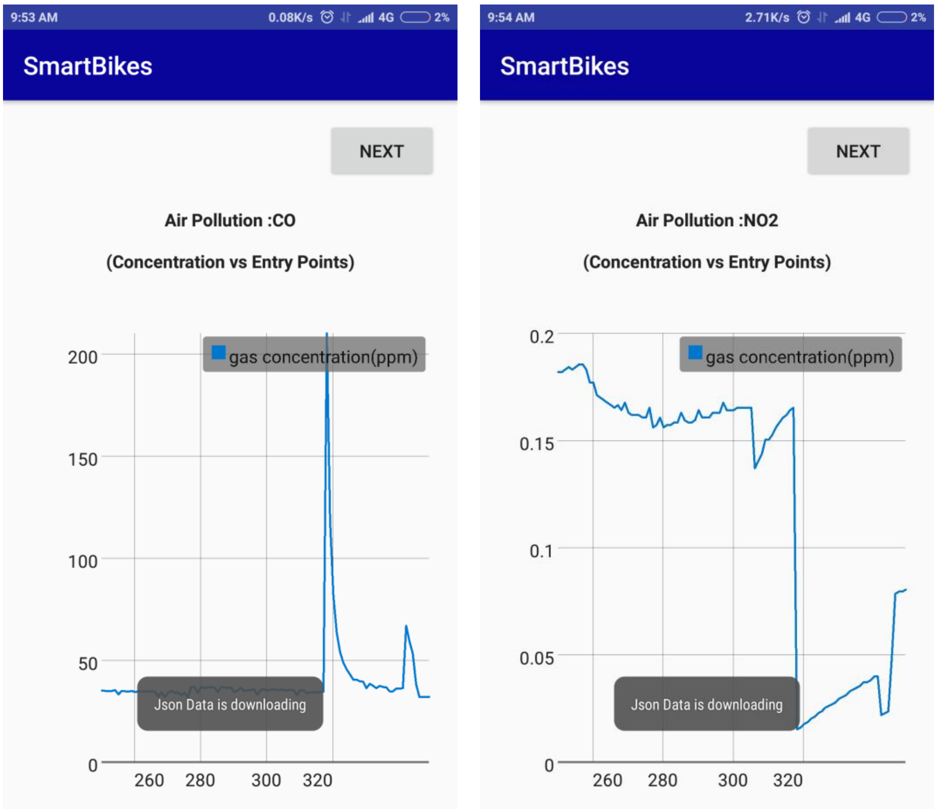


Figure 4.10 Graphs displaying gas concentration

The values are measured in real time are ppm values which are converted to AQI using the above formula. Specific values can be viewed with the help of available zoom feature in the graph itself. All graphs can be used by clicking on the next button. On tapping on specific points toast message appears displaying the axis coordinates.

4.4.3.6 Login and Sign up Activity

The Signup activity helps the user to register for an account on Firebase. The login activity gets the email and password from the user as shown in Figure 4.11. All the login credentials are stored in Firebase. Wrong Authentication blocks the user from entering into the application.

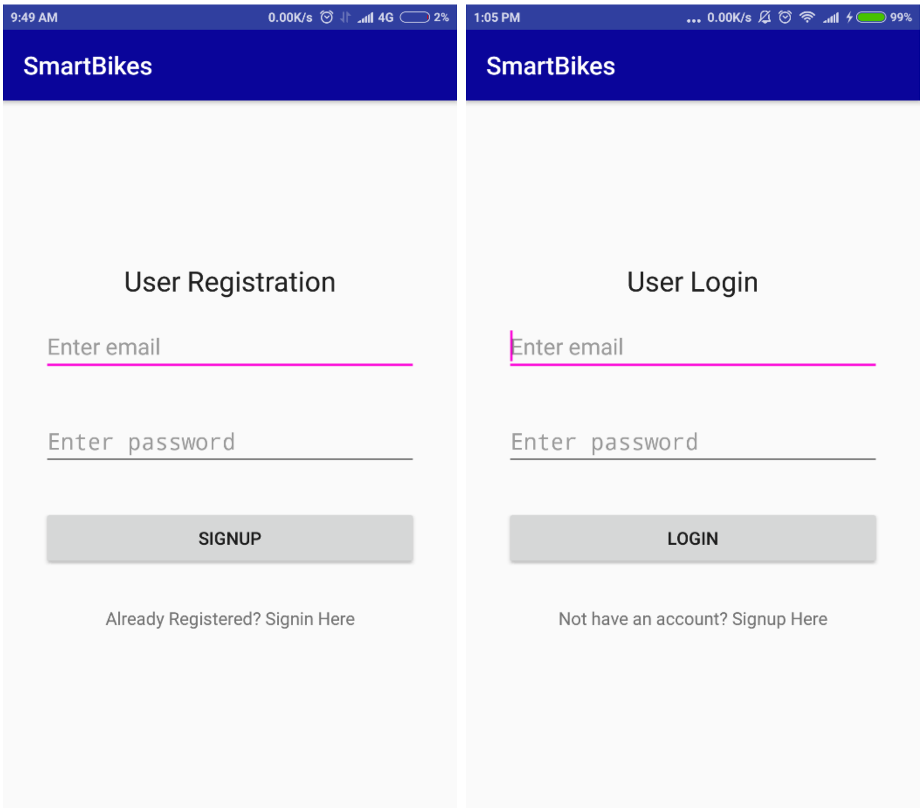


Figure 4.11 Login and Sign-up activity

4.5 FIREBASE

Firebase is a mobile and web application development platform with features and infrastructure required to build high quality applications. Firebase provides a plethora of features that help developers mix and match to fit their needs. It provides a database API which allows developers to store and sync data across multiple clients.

4.5.1 Firebase in Android Application

- A Device running Android 4.2.0
- Google Play Services SDK Repository
- Android Studio v1.5.0
- In order to manually add firebase to the app, it is necessary to possess a firebase console and create a firebase project by including the project name and the company package.

- By clicking ‘Add to your Android Application’, a Google-services.json file is provided which is to be downloaded and pasted into the app folder of Android Studio.
- Rules should then be added to the root level build.gradle file to include the google-services plugin.

```
buildscript {
    // ...
    dependencies {
        // ...
        classpath 'com.google.gms:google-services:3.0.0'
    }
}
```

- In the app-level gradle, the gradle-plugin should be enabled by adding the apply-plugin line.

```
apply plugin: 'com.android.application'

android {
 // ...
}

dependencies {
 // ...
  compile 'com.google.firebase:firebase-core:10.2.0'

  // Getting a "Could not find" error? Make sure you have
  // the latest Google Repository in the Android SDK manager
}

// ADD THIS AT THE BOTTOM
apply plugin: 'com.google.gms.google-services'
```

- The Firebase analytics can be finally included for the respective functionalities as follows:

Table 4.1 Firebase Functionalities

com.google.firebase:firebase-database:10.2.0	Real-time Database (Chat-Activity)
com.google.firebase:firebase-auth:10.2.0	Authentication (Authentication)

- The next step is to change the security rules to ‘true’ in order to enable writing into the database

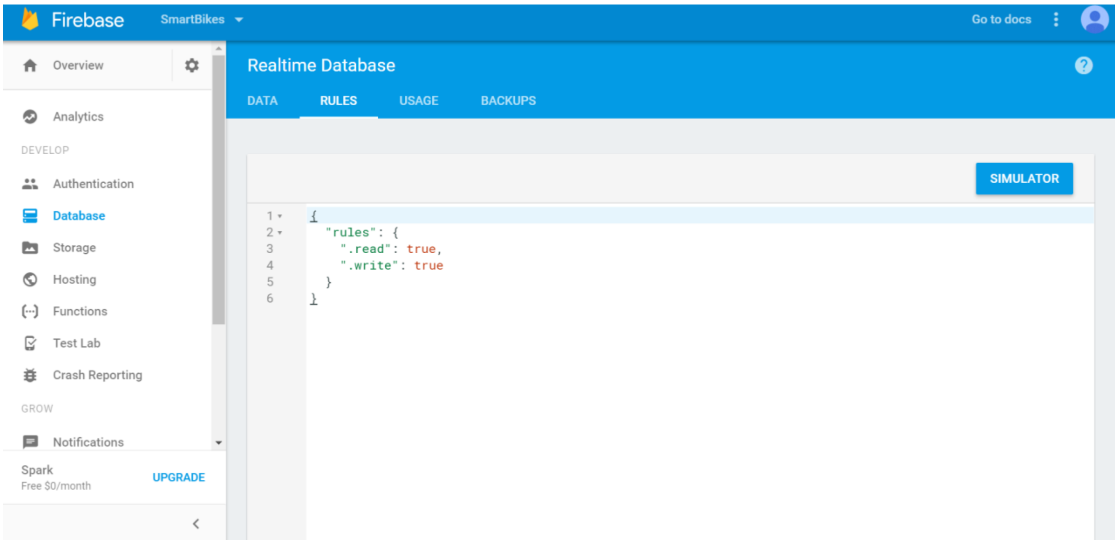


Figure 4.12 Firebase Security rules

By adding the above commands and by performing gradle sync, Firebase is integrated into Android application project.

Gradle Dependency Line	Service
com.google.firebase:firebase-core:10.2.1	Analytics
com.google.firebase:firebase-database:10.2.1	Realtime Database
com.google.firebase:firebase-storage:10.2.1	Storage
com.google.firebase:firebase-crash:10.2.1	Crash Reporting
com.google.firebase:firebase-auth:10.2.1	Authentication
com.google.firebase:firebase-messaging:10.2.1	Cloud Messaging and Notifications
com.google.firebase:firebase-config:10.2.1	Remote Config
com.google.firebase:firebase-invites:10.2.1	Invites and Dynamic Links
com.google.firebase:firebase-ads:10.2.1	AdMob
com.google.firebase:firebase-appindexing:10.2.1	App Indexing

Figure 4.13 Gradle dependencies

4.5.2 Real Time Database in Chat and Login

The Firebase Real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. When apps are built with Android, all of the users share one Real-time Database instance and automatically receive updates with the latest data.

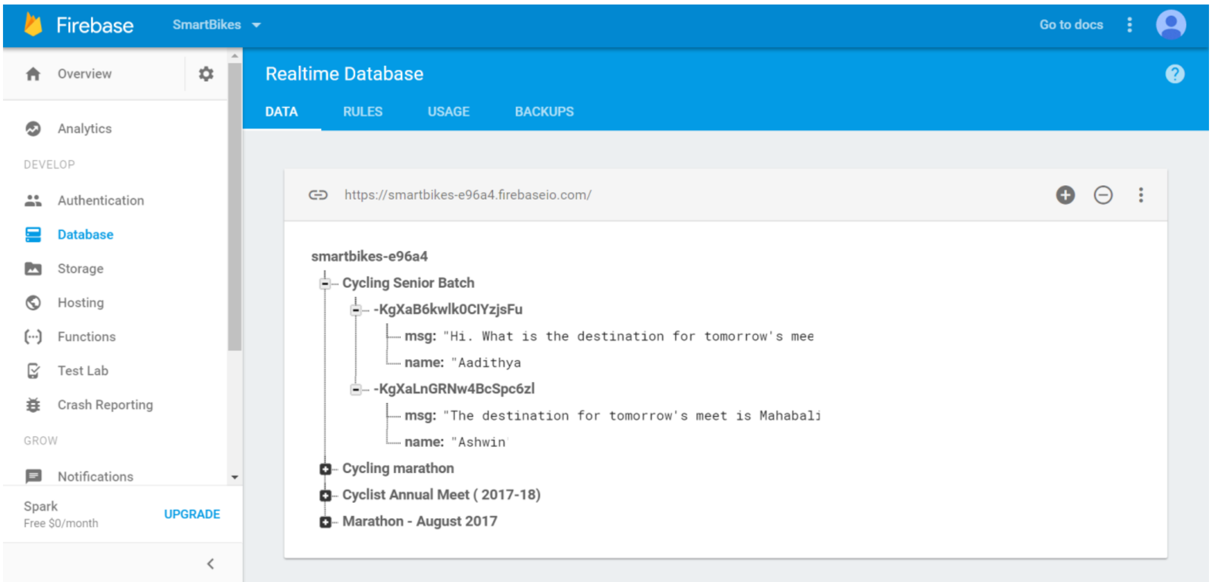


Figure 4.14 Firebase database chat room

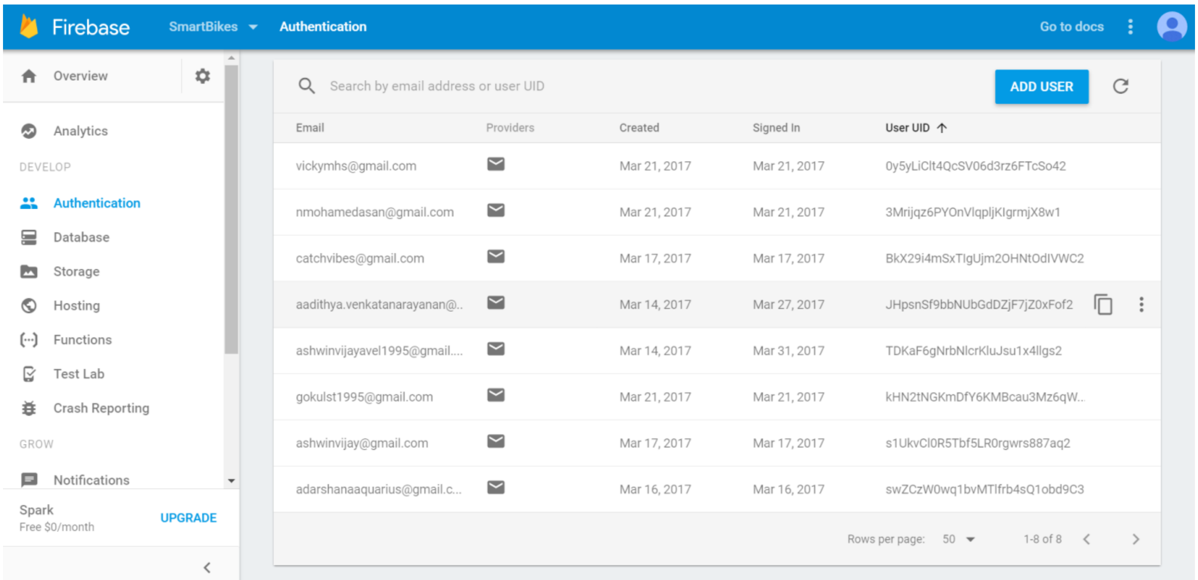


Figure 4.15 Firebase authentication database

4.6 SUMMARY

This chapter presented an overview about the software design and developmental tools used to create the android application. They are namely, Android studio and Firebase. The various activities that housed in the Smart bike app were discussed in detail. The application marks the front end of the system.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

To conclude, we have discussed, in this report, the detailed design and related algorithms for a project to make use of standalone system that helps both the producers and cyclists gain the maximum benefit. This project was successful at the attempts to solve the limitation that have not been addressed in a typical Smart bike monitoring system.

The air pollution monitoring and health monitoring was successfully implemented using sensors and a microcontroller. Location tracking and pollution monitoring was implemented using GSM/GPS module and sending to cloud using ESP8266 respectively. An Android application is developed to provide driver-assistance with the help of a mobile phone. The application houses various activities like login, Chat room, Graph and YouTube. The data thus obtained is used for smart city module and to promote bike sharing system.

The emphasis on use of smart system to collect environment variables as a part of Smart city initiative by Government of India is also addressed and justified. The potential of Smart bicycles and its impact on improving quality of transportation is immense. With rapid decline in the available resources such as fossil fuels, increase in traffic on the roads forces to switch to an alternative mode of transit. The untapped benefits include promoting bicycle sharing system where the citizens are encouraged to use cycles to commute in day to day lives. This opens the door to diverse possibilities, For instance, the accuracy of pollution mapping can be scaled up if the contribution of citizens is supplementary. Such system of data collection is rapid, reliable and economical.

Powering the system is one of the major areas of development especially in terms of research. Here the system is supported by 5000mAh power bank. The possible alternatives can be the addition of a dynamo or the use of solar cells.

Both are regenerative energy that can be produced while the cycle is in use. The accuracy of pollution mapping can be scaled up if the contribution of citizens is supplementary, thus making cycles a sustainable mode of transportation. The application endlessly evolves with the addition of activities to suit the need of users. For instance, detection of Carbon dioxide gas can be added to the circuitry. The health care monitoring can also include measurement of calories burnt. Other environment variables like Noise pollution monitoring is also a possible area of development. The system can be scaled down and made compact by integrating the system in a PCB.

A great impending market thrives in India thanks to ever increasing population. A modest initiative from the government to develop dedicated cycle lanes and traffic signals can put the Smart bicycles into immediate effect. With many countries like China and France already ranking low in most congested city list thanks to the use of cycles, which proves cycles are undoubtedly the most effective way of reducing traffic and improving the quality of life.

REFERENCES

1. Al-Ali A.R, Zualkernan I, Aloul F, "A mobile GPRS-sensors array for air Pollution monitoring", *IEEE Sensors Journal*, vol. 10, no. 10, pp. 1666-1671, 2010
2. Chris Kiefer, Frauke Behrendt, "Smart e-bicycle monitoring system: real-time open source and open hardware GPS assistance and sensor data for electrically-assisted bicycles," *IET on Intelligent Transportation Systems,* vol.10, no. 2, pp. 79-88, Mar 2016.
3. H. Johansson, P. Gustafsson, M. Henke, and M. Rosengren, "Impact of EcoDriving on emissions," *in Proc. Int. Sci. Symp. Transp. Air Pollut.* , Avignon, France, Jun 2003, pp. 97–105.
4. J. W. Kwon, Y. M. Park, S. J. Koo, and H. Kim, "Design of air pollution monitoring system using ZigBee networks for ubiquitous-city," *in Proc. Int. Conf. Convergence Information Technology*, 2007, pp. 1024–1031.
5. National Marine Electronics Association Data. [Online]. Available: http://www.gpsinformation.org/dale/nmea.htm
6. M.A. Razzaque, S. Clarke, "A security-aware safety management framework for IoT-integrated bikes," *in Proc. IEEE* 2*nd World Forum on Internet of Things* (*WF-IoT*), Dec 2015, pp. 92-97.
7. Smart E-bicycles monitor system (SEMS). [Online]. *Available*: *http://dx.doi. org/10.17033/DATA.00000016*, 2013
8. Soumya Kanti Datta, Rui Pedro Ferreira Da Costa, Jérôme Härri, Christian Bonnet, "Integrating Connected Vehicles in Internet of Things Ecosystems: Challenges and Solutions," *IEEE* 17*th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016, pp.1-6.

9. Stewart A Birrell, Mark Fowkes, Paul A Jennings, “Effect of Using an In-Vehicle Smart Driving Aid on Real-World Driver Performance,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp.1801-1810, Aug 2014.