IOT Based Smart Inventory Assistor

A Project Report submitted in partial fulfilment
Of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
In

Instrumentation and Control Engineering

Submitted by

Neil N Kadam
Reg. No.:160921202

Abhishek N
Reg. No.:160921298

Under the guidance of

Mr C. R. Srinivasan
Assistant Professor (Sr.)
Instrumentation and Control

DEPARTMENT OF INSTRUMENTATION AND CONTROL ENGINEERING

DEPARTMENT OF INSTRUMENTATION AND CONTROL ENGINEERING

MANIPAL INSTITUTE OF TECHNOLOGY
(A Constituent Institution of Manipal Academy of Higher Education)
MANIPAL – 576 104 (KARNATAKA), INDIA

Manipal
< Date >

CERTIFICATE

This is to certify that the project titled **IOT Based Smart Inventory Assistor** is a record of the bonafide work done by **Neil N Kadam**(*Reg. No. 160921202*) **& Abhishek N** (*Reg. No. 160921298*) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (BTech) in **INSTRUMENTATION AND CONTROL ENGINEERING** of Manipal Institute of Technology, Manipal, Karnataka, (A Constituent Institution of Manipal Academy of Higher Education), during the academic year 2019-2020.

Mr C. R. Srinivasan
Assistant Professor (Sr.)
Project Guide

Prof. Dr. Shreesha C
HOD, ICE.
M.I.T, MANIPAL

ACKNOWLEDGMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them.
We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude I give to our final year project Guide, Mr C. R. Srinivasan, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report.
We would also like to express our deepest gratitude to the Director, Dr. D Srikanth Rao, and the HOD of ICE, Dr. Shreesha Chokkadi, for providing us the opportunity to work on this project.
Lastly, we would like to express our gratitude towards the lab in charge and the faculty members of ICE for aiding us in the project and for their kind co-operation and encouragement which help us in completion of this project.

ABSTRACT

Inventory management is a challenging problem area in supply chain management. Companies need to have inventories in warehouses in order to fulfil customer demand, meanwhile these inventories have holding costs and this is frozen fund that can be lost. Therefore, the task of inventory management is to find the quantity of inventories that will fulfil the demand, avoiding overstocks. This paper presents a case study for the assembling company on inventory management. It is proposed to use inventory management in order to decrease stock levels and to apply an agent system for automation of inventory management processes.

In the present scenario number of inventory tracking systems and used in different types of applications. The challenge arise when we need to generalize the inventory and the stacking system and visualize it over a cloud based database system that will store and update data as per the values present in the storage, at last we outline how we can use different sensors and other methods to keep track of the quantity of the inventory. And then visualizing the data on a portable device, in an orderly format, to organize the system.

Inventory Management System also helps track theft of retail merchandise, providing valuable information about store profits and the need for theft-prevention systems. Automated Inventory Management System work by scanning a barcode either on the item. A barcode scanner is used to read the barcode, and the information encoded by the barcode is read by the machine. This information is then tracked by a central computer system. For example, a purchase order may contain a list of items to be pulled for packing and shipping. The Inventory Management System can serve a variety of functions in this case. It can help a worker locate the items on the order list in the warehouse, it can encode shipping information like tracking numbers and delivery addresses, and it can remove these purchased items from the inventory tally to keep an accurate count of in-stock items. All of this data works in tandem to provide businesses with real-time inventory tracking information. Inventory Management System make it simple to locate and analyse inventory information in real-time with a simple database search.

Therefore, the task of inventory management is to find the quantity of inventories that will fulfil the demand, avoiding overstocks. This paper presents a case study for the assembling company on inventory management. It is proposed to use inventory management in order to decrease stock levels and to apply an agent system for automation of inventory management processes.

LIST OF FIGURES

Contents

CHAPTER 1
INTRODUCTION

This chapter should include

- Project Work schedule

In this chapter we will have a brief discussion about the field this project is associated with and its general area of work present in the present day scenario and its applications. We also discuss the shortcomings of the current applied version of this project and its importance and how implementing this project will help in the improvement of the end result. We will also talk about the objectives of this project and the work schedule that was applied.

Inventory management is the supervision of non-capitalized assets, or inventory, and stock items. As a component of supply chain management, inventory management supervises the flow of goods from manufacturers to warehouses and from these facilities to point of sale. A key function of inventory management is to keep a detailed record of each new or returned product as it enters or leaves a warehouse or point of sale.

Organizations from small to large businesses can make use of inventory management to manage their flow of goods. There are numerous inventory management techniques, and using the correct one can lead to providing the correct goods, at the correct amount, place and time.
Inventory control is a separate area of inventory management that is concerned with minimizing the total cost of inventory while maximizing the ability to provide customers with products in a timely manner. In some countries, the two terms are used as synonyms.

With these systems, the procedures of inventory management extend beyond basic reordering and stock monitoring to encompass everything from end-to-end production and business management to lead time and demand forecasting to metrics, reports, and even accounting.
Some companies may opt to scan in inventory via a barcode scanner to increase efficiency along pick routes and accuracy.

Unlike an ERP system, an inventory management system focuses on one supply chain process. They often come with the ability to integrate with other software systems – point of sale, channel management, shipping – so you can build a personalized integration stack to the needs of your unique business.

Out of stocks and overstocks occur when a company uses manual methods to place orders without having a full grasp on the state of their inventory. This is a not a good predictor for inventory forecasting and results in too much stock or too little.

All of these mistakes will not only cost you money, but also cost you in wasted labor spent correcting the mistakes later. When you don’t implement management tools, your risk of human error mistakes goes up by the minute. And your customer reviews and loyalty take a negative hit as well.

Cloud-based inventory management is the monitoring and maintenance of a business's inventory levels using online software. Enabling businesses to avoid many of the errors and issues that arise with traditional methods of measuring stock levels, Cloud-based inventory management seamlessly keeps track of inventory coming in and going out of your business.

Traditionally, inventory management has been one of the most time-consuming and least popular tasks carried out by ecommerce businesses to keep their business running. It can be painstakingly boring and diverts resources from more important facets of the business, like growing the brand, improving efficiencies and adding new product lines.

By contrast, Cloud-based inventory management performs the same function, but with greater efficiency and effectiveness. As well as giving businesses back their time and resources; Cloud-based stock management systems reduce the likelihood of human error.

When it comes to inventory management, it pays to be precise. Stock is the lifeblood of your business so maintaining the right stock levels is crucial to your operations. Making the shift to a Cloud-based inventory management system is important for several reasons:
Instant stock level updates.

A cloud-based system can accurately inform you how much stock you have on hand. Cloud-based inventory management software can tell you whether you have enough stock for your demand forecast or whether you need to order more. This is an essential metric for customer satisfaction so you can avoid stock outs.

Crucially, Cloud-based inventory management reduces errors and the chance of 'stock walking'. The automated nature of the software means that inevitable human errors are eliminated. With Cloud-based apps, every stakeholder, from the manufacturer to the shipping company to the courier gets up-to-date information instantly, with no risk of duplications, missed orders or incorrect information, saving time and money.

A Cloud-based approach automates the supply chain so that the execution of an order – from stock reaching its reorder point to creating an order and sending it to the supplier – can be done without involving human labour, freeing your team's time for higher level decisions and priorities.

Safety of valuable information is guaranteed when Cloud-based inventory management is used: data backup on storage devices is not needed as all data is kept securely in the Cloud remotely. Any changes made are saved instantly, and security is taken care of by the service provider, who constantly update the Cloud inventory management software to avoid cyber-attacks.

Demand forecasting is one of the trickier aspects of running an ecommerce business: knowing how much stock to order in the future based on how much it's selling now is a fine art. Cloud-based inventory management systems analyse your data and create automated reports so you can make data-informed business decisions.

Finally, inventory management tells you a story about your business. Unless you have records of the amounts of products you've bought, amounts you've sold and amounts you hold, then you won't know what sells well, what doesn't and how you can change this to grow your

business. Cloud-based inventory management allows you to link stock levels with your ecommerce store, accounting software, 3PL provider and more, so that you're always aware of how you're doing.

As well as saving you time, Cloud-based inventory management software effectively manages your inventory to reduce errors and prevent stock from going missing. Moving away from spreadsheets and the errors that come with manual data entry, stock management software measures how inventory moves through your overall business. All products and variants have an audit trail activity feed, so you know exactly what products went where, and when.

The main objective of inventory management is to maintain inventory at appropriate level to avoid excessive or shortage of inventory because both the cases are undesirable for business. Thus, management is faced with the following conflicting objectives:

1. to keep inventory at sufficiently high level to perform production and sales activities smoothly.

2. To minimize investment in inventory at minimum level to maximize profitability.

Cloud-based inventory management offers a compelling alternative to manual approaches to inventory management or costly on-premise ERP. The best cloud system provides real-time visibility into inventory, with anywhere, anytime access to critical information. It can function at the core of an ERP system, integrating seamlessly with demand planning, financials and logistics. Automated capabilities eliminate manual inputs while maximizing efficiency throughout the inventory lifecycle.

Scalability, flexibility and visibility are the three key elements that comprise an integrated approach to inventory management in the cloud.

The right cloud inventory management system will scale as your business grows, from a small single-warehouse operation to an international distributor spanning multiple markets. The cloud makes it easy to add users, functionality, warehouses and suppliers—without the large-scale cost and effort required to implement a new on-premise system, or install it in new offices.

A cloud system lets you select the right level of inventory management sophistication for your business or industry. It lets you use only the functionality you need, without needless complexity. As your business grows, cloud inventory management scales with you by offering more sophisticated features and virtually unlimited capacity for more users and information.

These changes could involve simply a modification in reporting to improve inventory management speed and precision. Or they could be changes to how you see information, or deliver data to select managers based on their job function or location. Change could mean adding a new shipper or bar-code system, or introducing capabilities for quantity-based pricing, lot and bin management or landed cost calculations.

A cloud-based inventory management system built with flexibility in mind allows you to define your business processes into the system. It offers flexibility for you to implement

customizations and business rules that support your unique requirements. The optimal cloud architecture preserves these customizations for you, so that they remain intact even as the cloud provider periodically upgrades your system to the latest release.

Unlike spreadsheets or desktop applications, the cloud doesn't tether you to a desk. Web-based inventory management gives your personnel anywhere, anytime access to real-time data, whether on the road with a mobile device or receiving alerts on the warehouse floor. Metrics-driven dashboards let your managers continuously monitor and improve performance.

The cloud takes inventory management to a new level. It gives you the ability to easily grow with your business. It allows you to configure the system to your business requirements. And it gives you full visibility within your warehouse and across multiple locations.

CHAPTER 2
BACKGROUND THEORY

In this chapter we will discuss the specifics of the project and the literature review that was done. We will realize the present state and mention a brief background theory. Towards the end of this chapter we will summarize the reviews and theoretical discussions along with general analysis.

Versatile application advancement as of late is developing exponentially. Today every single individual in this world has an advanced mobile phone in his pocket. Cell phones consolidate a scope of capacities, for example, media players, camera and GPS with cutting edge processing capacities and contact screens are getting a charge out of consistently expanding prevalence Cell phone's assistance us to accomplish a scope of undertakings through something known as applications or Apps to short. As indicated by Gartner [3], Google's Android, Apple's iOS and RIM's Blackberry all have something like a 10 percent piece of the pie. For finishing this survey paper and learn about this subject an aggregate of four research papers were utilized which comprehended reasonable and ebb and flow situation of Cross-stage versatile application advancement[1].

Our communication application or messaging system aims to provide a platform for two individual users separated by a certain geographical distance to communicate with each other, through the Internet with the help of various Firebase tools. The system is divided into various modules, each module dealing a separate task and working in sync with the other modules without any conflicts [2].

Examination of most recent cross-stage mobile application improvement approaches which are at present accessible in the market. A portion of the cross-stage versatile application advancement approaches are Phone Gap, Titanium and so on. The recognize approaches that utilize a run-time condition and those that create stage explicit applications from a typical code base at incorporate time. The last mentioned, generator-based classification incorporates display driven arrangements and cross-gathering [1].

IoT- primarily based pollution watching and prognostication System: This paper [1] introduces IoT with environmental protection and it puts forward a form of pollution watching and prognostication system. Wireless device network for period pollution monitoring this paper [2] presents the system which consists of many distributed watching stations that communicates wirelessly with a back-end server victimisation machine-to-machine (M2M) communication [3].

The server used for Android apps are Oracle SQL, Microsoft SQL Server, and MySQL which are connected to the server with PHP files. Then Firebase came into existence for Android apps which uses JSON for storing data. The other servers use a table (rows and columns) format for storing data. Firebase is NoSQL based. There are very few cloud based server available which are similar to firebase, like: AWS Mobile Hub- It is integrated console that helps to create, build, test, and monitor the mobile apps that leverages AWS services [4].

The balance of these goals varies, but they are always there, whether it’s a connected industry in automotive, cloud services for control of consumer goods, or smart metering and smart

grid architectures in utilities [8]. Implementation of a wireless automatic meter reading system (WAMRS) which incorporates the widely used GSM/GPRS network. The system includes a microcontroller, which periodically transmits power consumption values calculated from the sensed voltage and current values via an existing GSM/GPRS network, to a master station. The main disadvantage of this technology is distance factor [9]. A strong GPRS or a GSM network coverage at long distances may not be available whereas the other disadvantage might be speed of operation [5].

The Cloud has become an important factor in enterprise computing and will remain so in the foreseeable future. According to [2], 83% of enterprise workloads will be in the Cloud by 2020. Hence, there's both a need and an opportunity to provide educational training to equip students with the necessary skills to compete in this high-demand market. Teaching Cloud computing concepts covering the full-stack business solution requires introducing students to the different full-stack layers [6].

These projects play an important role in our daily life, especially after the emergence of new technologies that have made these projects more flexible and close to the needs of the human community and also when cloud computing started to provide services to these projects. In this paper I shed light on one of those techniques provided by the multinational technology company (Google), which is a (firebase) technology that provides many services for projects whether small project or large, including database service, license service and notification service etc., In this paper I will provide a detailed explanation of notification service over IoT. This work will implement in two parts: Hardware which consist of Arduino, an ESP8266 Wi-Fi controller, and a keypad that use to access to the system when typing the secret code, and software include android application that contact with firebase.

When a new user turn on the device, it will send notify to the smart phone using firebase notification to alert someone tried to access to the system [7]. Home automation can be defined as it is the removal of human interaction [5]. Using the concept of IOT we make sensors to communicate with each other which are powerful in automation [7]. The efficient use of electric energy is highly dependent on energy metering. Presents an idea or a concept for smart home. Smart home or home automation can be said as the residential extension, it also involves the automation and controlling of lightings, ACs and security, which includes other home appliances. Data is collected with the help of node MCU ESP8266 and Firebase [4].

The most noticeable mixture structure till date for cross stage application improvement is Phone Gap. Telephone Gap was initially made by Nitobi Soft-product, which has been procured by Adobe. The improvement presently happens in the Apache Cordova undertaking of the Apache Foundation, of which Phone Gap is a dispersion. Telephone Gap a well-known cross-stage versatile application improvement apparatus is inexactly founded on jQuery which is an exceptionally quick library change device [1].

Many developers are currently developing messaging applications with online solutions similar to Firebase, which provide real-time database integration facilities. Various open-source platforms such as Parse Server or Horizon offer similar services such as Firebase and offer developers to migrate from one vendor to another, but they also come with their own problems. Developers are also trying to develop methods to optimize file transfer through such applications and to integrate more technologically advanced features into their applications [2].

The [7] has an occasional price geo-referenced air-pollution measuring system is used as early warning tool: the full system is Connected to an inexpensive board with inherent Wi-Fi permitting to send the info to the IoT cloud in period victimisation MQTT protocol, associate degreed so the georeferenced knowledge may be printed on an open access platform using IOT[3].

Firebase is a mobile and web application development platform enveloped by Firebase, It help to build easy mobile app, it is easy to authenticate, user friendly cloud [4].

In addition, the ACM/IEEE Computer Science Curricula 2013 (CS2013) added platform-based development (PBD) as a new knowledge area to the Computer Science body of knowledge. CS2013 recommended adding web application development and applying it over a wide range of ecosystems as part of the PBD knowledge area. CS2013 acknowledged the “increasing use of platform-specific programming environments, both at the introductory level and in upper-level electives” [1].

Thus accidentally tripping or running into thing is decreased. Home automation system using IOT is a way through which one can explore and control home appliances or devices with the help of internet connection. Internet has become the basic need for everyone and thus internet can be used for controlling the basic devices or appliances like TV, lights, fans, Air conditioners and much more.

This technology is operated with the help of smart phone and sensor (The only thing which is needed is the automation system should be connected to the cell phone) [8].

CHAPTER 3
METHODOLOGY

In this chapter we are going to talk about the development process of the project and explain the steps taken towards the management and design of the project. The project was divided into several parts depending on the type of work required and was implemented and designed in a similar fashion. The flowchart below explains the progress steps that were taken:

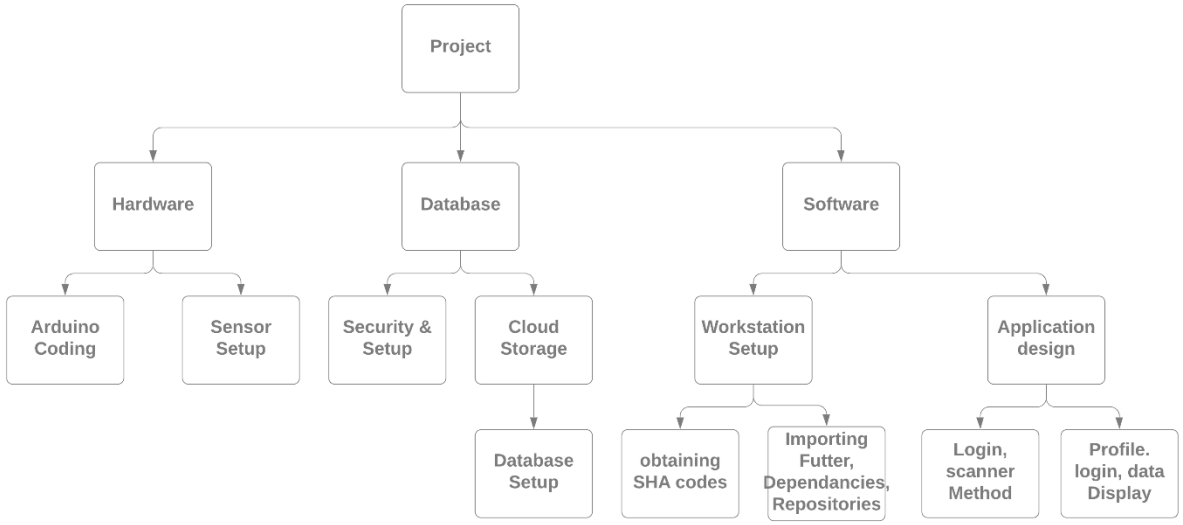


Fig. 1: General workflow

The main development method which was applied in this was using different container elements and then measuring the same using the setup containing the ultrasonic sensors.

The .json declaration has been used for online server communication, because of which the communication with the online server is possible [11] [13].

The development of the project was organized and approached in three separate phases and were executed in steps corresponding to those specific phases.

3.1 Hardware phase

The Arduino Yun is a hybrid Arduino consisting of ATmega32u4 and Atheros AR9331 Thus it supports a Linux distribution based on Openwork named Linino OS. Since we are using the Arduino Yun, this enables us to use the Wi-Fi feature to enable the wireless communication btw the Arduino and the IDE software on a system, thus allowing to update the software on the Yun remotely without having to connect the Arduino to the system. This Ultrasonic sensor are connected to the Arduino Yun which are used as the sensors to collect information from the containers.

This sensor information is sent to the online server i.e. the Firebase. Furthermore, the information is sent to the online server in the form of percentage, where the container size is taken into

consideration and the fill rate is converted to its percentage format, which results in easier display of the information in percentage format and in a donut graph.

3.1.1 Coding for the Arduino

The initial coding part involves setting up the Arduino Yun for Wi-Fi communication, where the password for the Arduino can be set thus providing security for accessing and altering the code [9] [12]. The Arduino is set up with the different sensors such as the HC-SR04 which was used in this experiment as one of the sensors to measure the content present in the container [10].

Below is the code for the initialization of the pins for sensors and the coding for calculation of the distance based on the time of traveling of the ultrasonic wave which is converted to centimetre format. Since in this project we have considered a container having a depth of 10cm the coding is done to get the percentage filled for a 10cm container.

Initialization done for '.jason' and the Ultrasonic Sensor:

```
#include <Bridge.h>   //bridge library - for bridge communication between the server and Arduino
#include <Process.h>   //process library - for the online communication
#include <Console.h> //console library - Wi-Fi communication

Const char *FIREBASE = "https://fir-ee534.firebaseio.com/"; // declaring firebase server
Const char *NODE = ".json"; //'json declaration for online server communication extension

Const int UPDATE_RATE = 1000; //update delay rate
Const int trigPin3 = 6; //trigger pin used in the Arduino to connect the US sensor
Const int echoPin3 = 11; //echo pin connection
```

Ultrasonic sensor coding and conversion to centimetres

```
PinMode (trigPin3, OUTPUT); // Sets the trigPin as an Output
PinMode (echoPin3, INPUT);    // Sets the echoPin as an Input

// Clears the trigPin
DigitalWrite (trigPin3, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin3, HIGH);
delayMicroseconds (10);
digitalWrite (trigPin3, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
long duration3 = pulseIn (echoPin3, HIGH);
// calculating the distance
Double distance3= duration3*0.034/2;
```

Updating Information on the cloud server

```
//Call used in main function
// Update value on Firebase
      request (distance3);
      wait ();
      response ();
// defining the function outside the main
void request (double distance3) {
char per3 [10];
dtostrf (distance3, 3, 2, per3); // distance is converted to string
// Build curl command line
  // Includes HTTPS support
  // PATCH
  // Single value so no JSON
  process.begin (“curl”);
  process.addParameter (“-k”);
  process.addParameter (“-X”);
  process.addParameter (“PATCH”);
  // Body
  // Value to update
  process.addParameter (“-d”);
  Sprintf (
    Buffer,
    "{\"distance1\": %s}",
    per3
   );
  process.addParameter (buffer);

  // URI
  Sprintf (
    Buffer,
    "%s%s",
    FIREBASE,
    NODE
  );
  process.addParameter (buffer);

  // Run the command
  // Synchronous
  Process. Run ();
}

Void response ()
{
  Bool print = true;
  Char c;

  // While there is data to read
  while (process.available ())
```

```
  {
    // Get character
    c = process.read();

    // Debugging
    #ifdef DEBUG
      Console.print( c );
    #endif
  }
}

void wait()
{
  // periodically check curl process
  while( !process.available() )
  {
    delay( 100 );
  }
}
```

The above code represents the setting for one Ultrasonic sensor, in the experiment 3 such Ultrasonic sensors were used for different purposes,
The setup for the same is shown in the figure below

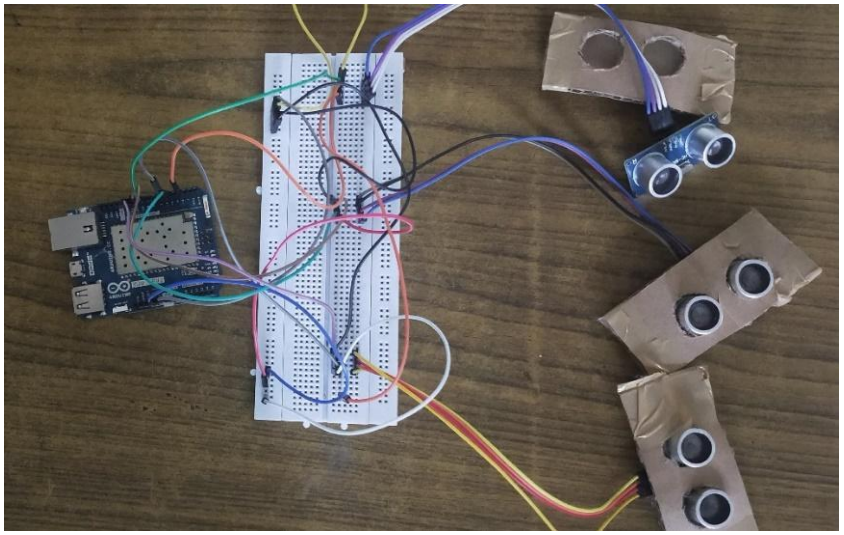


Fig. 2: Final Hardware Image

3.2. Database Creation and Management

To reduce the load on the mobile application development we choose Firebase as a Mobile Backend. Firebase is a complete Package of products that allows one to build web and mobile apps. Firebase offers MBaaS or Mobile Backend as a Service and offers a method to link web and mobile applications to a cloud based backend storage and API.

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of March 2020, the Firebase platform has 19 products, which are used by more than 1.5 million apps. It helps the developers to build real-time applications for IOT.which is one of the mobile platform by sending messages and notifications[3]. Firebase is a free provider for low scale projects but can be easily scaled to an industrial level at which point it will be charged, but as compared to the scale of the network the cost is almost insignificant. Real-time Database is a cloud-hosted database.

Data is stored as JSON and synchronized continuously to each associated client. When any cross-platform application is developed with iOS, Android, and JavaScript SDKs, the greater part of the user's demand is based on one Real-time database instance and this instance gets updated with each new data[2]. Firebase is NoSQL based. There are very few cloud based server available which are similar to firebase [4]. Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of March 2020, the Firebase platform has 19 products, which are used by more than 1.5 million apps.

3.2.1 Introduction

Firebase is a mobile and web application development platform enveloped by Firebase, It help to build easy mobile app, it is easy to authenticate, user friendly cloud [5]. The firebase provides a service called BAAS. The BaaS service model focuses on the business logic of mobile applications, thereby freeing the app from needing to perform long-running computational tasks [6]. To enable and create a working firebase database we need to choose between two database options those are the Real-time Database and the Cloud Firestore.

The Rea-time Database stores and syncs data in real-time across all connected devices whereas the newer version of the real-time database i.e. the Cloud Firestore provides real-time updates, powerful queries and automatic scaling for your application. As this project is a small scale demonstration of the data management system we will be using the Real time database. It stores the data in JavaScript Object Notation (JSON) format which doesn't use query for inserting, updating, deleting or adding data to it. It is the backend of a system that is used as a database for storing data [4].

3.2.2 Security and Setup

To begin creating a backend database we first need to create the project. We will use the default firebase account and this will provide us with a project ID. This ID is important as it will define the location of the storage of the project. After the project has been created on the firebase account we need to set up an Authentication method. It supports authentication using passwords, email id or username, phone number, etc. Users can be allowed to sign in to a Firebase app either by using FirebaseUI as a complete drop-in authentication solution or by using the SDK to manually integrate one or a few sign-in techniques [2].

This will allows us to Authenticate and Manage users from a variety of providers without server side code. This Authentication is only for developers. It is a service that can authenticate users using only client-side code and it is a paid service. It also includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase [4]. This will allow us to make a hierarchy and will allow us to manage different development teams to work on the project. This will also allow us to give permissions to read and write and edit security code form the server end. The user has to input the required credentials and the Firebase system checks whether these credentials are correct or not [2].

When a new user turn on the device, it will send notify to the smart phone using firebase notification to alert someone tried to access to the system. The owner will resend notify to device to give permission and store it in firebase database [7].

3.2.3 Cloud Storage

The next step would be to set up the Cloud Storage. We need to cloud storage to upload and download files directly from clients. Again for the Firebase SDK and the cloud storage to work seamlessly we need to provide a declarative security language that lets you set access controls on individual files or group of files, such that one can make files public or private.

Firebase Storage was designed for application developers who need to store and serve user-generated content, for example photos or any other file. It is backed by Google Cloud Storage which is cost-effective object storage service [4]. It gives secure document transfers and download for Firebase applications, regardless of network quality.

Firebase Storage is upheld by Google Cloud Storage, a capable, basic and cost-effective object storage service [2]. And because of cloud storage we need not migrate to any other provider as the storage scales automatically.

3.2.4Setting up the Database

After this we need to set up the database. In real-time database we need to edit the default rules to suit our needs. These rules as mentioned before give permissions to developers to read and write. An API is provided to the application developer which allows application data to be synchronized across clients and stored on Firebase's cloud [4]. Lastly in the usage tab we can observe the usage statistics of the users and provide modifications when needed. The data stored is highly secured and is robust in nature means it resumes from the last point if any network error occurs [4].
After setting up everything we are done with the online database Phase for now. The connection of the application to the database will be covered in the next section.

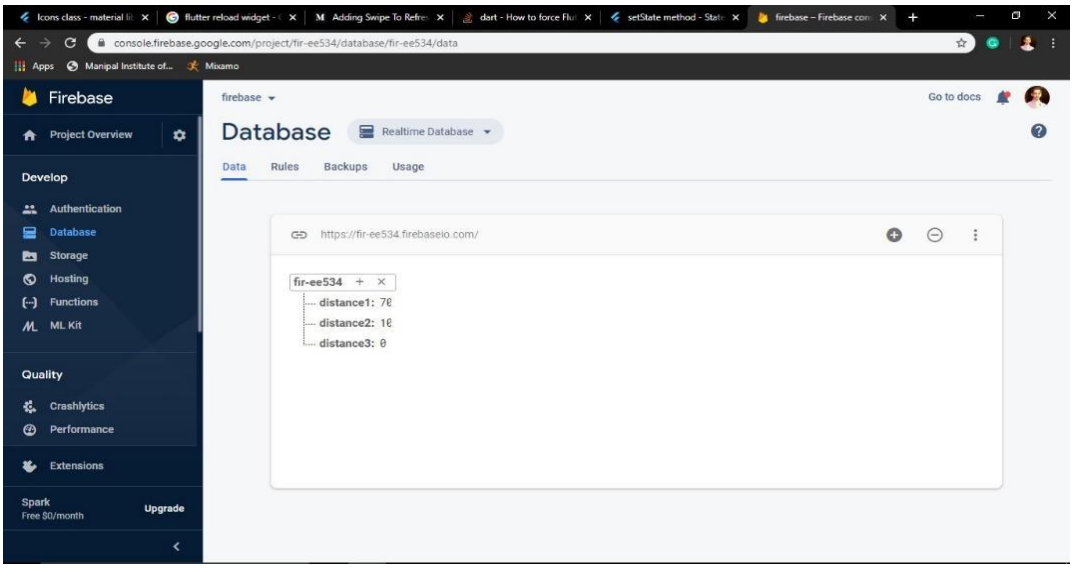


Fig 3 : Firebase Database Datapage

3.3 Mobile Application Development

For the development of the mobile application that can be very easily accessible as it would be based upon android OS. We will be using the Android Studio IDE for the development. We choose the android OS as it is more widely used and very easily accessible on various devices. It is also very easy to code, manage and update.

It is also important to have JAVA installed on your development computer as it is required in some phases that will be discussed later in the upcoming parts. Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Windows, Mac, Linux, Google Fuchsia and the web. The first version of Flutter was known as codename "Sky" and ran on the Android operating system.

3.3.1 Obtaining the SHA-1/256 Keys

First, we need to locate the SHA-1 and SHA-256 keys that are present in your java directory. SHA-1 is a unique key generated for your PC that can be used for signing/ it is mainly used for submitting the Google API in case of android. Every system with an active JDK installed on it will have a unique SHA-1 and SHA-256 key present. To obtain these keys we need to find the JDK directory in our system and open command prompt in that directory space. After doing so we need to input the following command:

key tool -exportcert -alias androiddebugkey -keystore [PATH_TO_.DIRECTORY] –listv

Upon replacing [PATH_TO_DIRECTORY] with the path present in your system we should be able to run the terminal command. In some cases you will be prompted with a password which you can enter (usually android) after which you the list of certificates will be printed on the screen. After this you can simply copy and paste the SHA-1 and SHA-256 keys in a separate word document or a notepad file

Next we need to specify an app file on the firebase page. This will be the link between our android application and our Firebase database. After making the app on firebase we provide a package name and note this package name in a doc as we will need to use the same package name for the application as well. Once we are done we will be required to download and save the **google-services.json** file in the root folder of your application development directory.

3.3.2 Importing Flutter

Now we need to import flutter into our Android Studio IDE after downloading the required files from the flutter website. Flutter is an open Source UI development software that is used to develop front end of application [1]. As we already have a backend due to the firebase database we are not required to specifically create a backend in the application so we can prevent a lot of complications in this phase.

Now we need to add the required repositories and dependencies so that we can use the required functions and processes. Flutter has developed different dependencies that reduce the workload of generating extended process by just probating them as simple functions. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it [8]. Then all we need to do is just call these functions to use the embedded processes.

This makes the length of the code much shorter and is also much more helpful in debugging and post processing of the code. As the processes and embedded into the required functions they can be called repeatedly without the need to write the entire process over again. This again further simplifies the code although the processing is a bit more tedious, but due to our app being only front end it will not affect a lot.

3.3.3 Repositories and Dependencies used

The following repositories were used in the project:

1. **cupertino_icons: ^0.1.2** – This is a repository which contains the default set of icons and assets that are required to make the user interface for the application.
2. **firebase_auth: ^0.15.4** – This is a futter plugin to use the Firebase Authenticaton API.
3. **firebase_database: ^3.1.1 –** This is a flutter pugin to use the Firebase Realtime Database API.
4. **google_sign_in: ^4.1.1 –** This is a flutter plugin for Google Sign In.
5. **provider: ^4.0.4 –**
6. **firebase_core: ^0.4.4 –** This plugin enables the Flutter plugin to uses the Firebase Core API, which allows connections to multiple firebase applications.
7. **firebase:^7.2.1 –** This package provides three libraries :
   -package:firebase/firebase.dart
   - package:firebase/firestore.dart
   - package:firebase/firebase_io.dart
8. **intl: ^0.15.8 –** This package provides the internationalization and localization facilities like message translation, plurals and genders, date/time/number formatting and so on.
9. **percent_indicator: ^2.1.1** – This Package provides us with circular and linear percent indicators as a function.
10. **qr_flutter: ^3.2.0 –** This is a simple library for simple and fast QR rendering via a widget/custom painter.
11. **barcode_scan: ^2.0.1 –** Another simple library for barcode rendering.
12. **path_provider: ^1.6.4 –** Flutter plugin for locating commonly used locations on the filesystem on android as well as IOS.

The following dependencies were used in the project:

1. **org.jetbrains.kotlin:kotlin-stdlib-jdk7**
2. **com.google.firebase:firebase-database:19.2.1**
3. **com.google.firebase:firebase-database**
4. **junit:junit:4.12**
5. **androidx.test:runner:1.1.1**
6. **androidx.test.espresso:espresso-core:3.1.1**
7. **com.google.gms.google-services**

3.3.4 Designing the Application

Now the application needs to have multiple pages to organize the different tasks. To simplify the process we will provide a separate page for each task. For every page that we need we will make a different code file and then piece them all together by calling them into a single main file. If we need to make a common function then we will create another file to develop the function.

Keeping the function in separate files will allow us to call them when required without editing any variables and will also allow us to have a cleaner code and easier debugging. In case of running the application in debug mode we need a device to run the application on.

Although we can emulate a device on the development PC itself it is better to have an external device, this will make the processing and the building of the application more efficient and faster. We can use external mobile phones running any recent version on android as an external device to run the application and test it.

3.3.5 Making a Login Method

To make the project more secure and to differentiate between several users we can simply make a login based user system that uses the users Google sign in on their phones to sign in to their application account. User will basically use their Google account to sign in.

For the sign in method we will make a separate script and then call it into the main script in the end while building the application.

In the sign in system we will have 2 separate methods consisting of sign in, sign out. Both these methods have a similar make and commands but their executions is mildly different.

For the system we are required to use two packages:

1. package:firebase_auth/firebase_auth.dart – This package has functions that will allow the online database to register and locate the specific user.
2. package:google_sign_in/google_sign_in.dart – This package has methods that allow the use of android Google based sign in via their email and allow us to register the used id and email.

To begin we will have to run the following command:

final FirebaseAuth _auth = FirebaseAuth.instance;

This will authenticate the firebase to allocate users according to their email IDs. As soon as this command is called the application will await for a response from the google sign in method to allocate an email ID to the available string present.

To initiate the google sign in we need to create a function like the following:

```
Future<String> signInWithGoogle() async {
  final GoogleSignInAccount =await googleSignIn.signIn();
                                    finalGoogleSignInAuthenticationgoogleSignInAuthentication=await
GoogleSignInAccount.authentication;
  final AuthCredential credential = GoogleAuthProvider.getCredential(
    accessToken: googleSignInAuthentication.accessToken,
    idToken: googleSignInAuthentication.idToken,);


assert(user.email != null);
  assert(user.displayName != null);
  assert(user.photoUrl != null);


  assert(!user.isAnonymous);
  assert(await user.getIdToken() != null);
```

The async statement allows the application to await reply from the user and the above statement will create a pop up that will allow the user to login to their gmail account. Once the user logs in to their account their email, display name and their profile photo is acquired and stored. It is checked that these are not null or empty.

In case it is empty the method will send the process back to the top at the beginning of the process. Once the sign in is complete the process will have a user ID token, which will be unique to each user. We will assign the profile details to separate strings which will allow us to use them as text files.

```
final FirebaseUser currentUser = await _auth.currentUser();
assert(user.uid == currentUser.uid);
```

The above statements will take the user ID and the ID token generated are set to the firebase statement which was awaiting before and will send the data to the database server.

The second method that we need to implicate is the sign out method. This is a fairly simple method as compared to the sign in method.

```
void signOutGoogle() async{await googleSignIn.signOut();
```

The above statement will simply sign you out and clear the profile strings and basically reset the application. This will create our login/logout method.

3.3.6 Making the Profile Page

For the profile page we will create a different dart file. For the profile page we will use the profile strings that we got during the login to display the user information on the profile screen. Flutter works in widgets, what this means is that everything in flutter starting from a picture to text everything. This makes planning and creating the general layout of the application significantly easier.

We will not discuss the planning of the application but rather we will talk about the important points like using the profile strings and connecting the sign out methods to the page and create a page switch in case of log out.

After the planning and the basic page layout is created we can begin developing the page specific parts.
First need to extract the image from the string. The image_URL string only provides us with a link to the image provided in the actual Gmail profile picture present in the users Gmail account.

We can access that image via the URL and then flutter will build the application while down loading the image from the Gmail server. We need to following command to make the display widget.

```
CircleAvatar(
        backgroundImage: NetworkImage(
          imageUrl,
        ),
        radius: 60,
        backgroundColor: Colors.transparent,
      ),
```

Next, we need to use the other profile strings specifically the user name and the email strings. We got the information during log in and saved it in strings. To use these string its fairly simple as they can be simply considered as text files.

This means we can create simple text widgets and just use the string and we will acquire the required output. We need the following command for that widget.

```
Text(
        name, //string name
        style: TextStyle(
          fontSize: 25,
          color: Colors.deepPurple,
          fontWeight: FontWeight.bold),
      ),

Text(
        email, //text email
        style: TextStyle(
          fontSize: 25,
          color: Colors.deepPurple,
          fontWeight: FontWeight.bold),
      ),
```

Fig 4 : Profile Page Design

Next we need to create the sign out option on the page so that the user can log out. To create the sign out button we need to create a button widget and give it and action effect and link it to the sign out method that we created before on the login method and then the application should push the user screen back to the login page We will use the following set of code for to create the button.

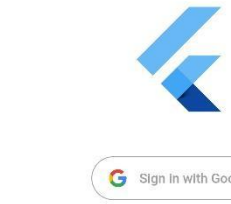
```
RaisedButton(
        onPressed: () {
signOutGoogle();
Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (context) {return
LoginPage();}), ModalRoute.withName('/'));
        },
```

Once we create this it will create a loop between the login and profile page. This will allow the user to login and logout safely and successfully and will allow the user to safely login and log out through the application itself and will not have to do it manually via their gmail. This finishes the Profile page

3.3.7 Login Page

We have created the methods for login in another dart file before so we will only have to call the method and design the UI for the login page the login page will be linked to the data page and the profile page has a push method. This page will not need many widgets, only the login option.

The page can have other widgets and options but for the simplicity of the app we will give it a simple UI.




```
Widget _signInButton() {
  return OutlineButton(
    splashColor: Colors.grey,
    onPressed: () {
      signInWithGoogle().whenComplete(() {
        Navigator.of(context).push(
          MaterialPageRoute(
            builder: (context) {
              return DataPage2();
            },
```

Fig 5 : Login Page Design

When the Log in method returns true the above method will push the page in view to the datapage. We will later add a side drawer which will send us to the profile page and hence the log out button. And in case the log in is a failure a simple message will pop up stating the same.

2.3.8 Data Page

For the data page we will have a decent UI made up of circular progress indicators and Bar Progress indicators. As a repository for these types of indicators is already available there is no need to create them from scratch using the paint creator methods. We can simply add the repository can call these functions to use them.

As this page will be a little more complex as compared to the other pages we will need to plan the design properly or else the system will not be able to compile the application due to the parameters not being aligned properly with the desired screen size if this occurs there will be a red screen showing the error code and description on the device. First we need to set up a basic layout of the page

```
return MaterialApp(home: Scaffold(
```

```
appBar:  AppBar(title: Text('Datapage'),),
body: ListView(
```

This will create a simple appbar on the top of the screen and will have datapage written on it and the body list view will make the contents of the body appear to be like vertical list[19].

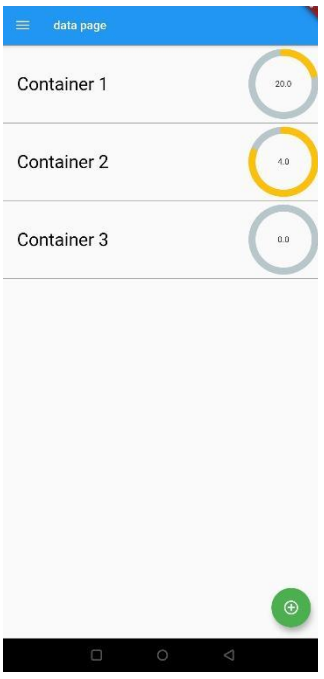
```
Container(
        height: 1,
        color: Colors.grey,
      ), //divider
```

We will use the above method as a divider of sorts to separate different widgets from mixing into each other and causing an error caused by lack of individual space.

Next we will create a slot in the listview discerning to a value from the firebase database.

```
Container(
        height: 110,
        child: Row(
          children:[

            Container(
              width: 350,
              padding: EdgeInsets.fromLTRB(20, 0, 0, 0),
              child: Text('Container 2',
                style: TextStyle(
                    color: Colors.black,
                    fontWeight: FontWeight.w400,
                    fontSize: 30
                ),
              )
            ),
            CircularPercentIndicator(
              progressColor: Colors.amber,
              percent: meso,
              animation: true,
              radius: 100,
```



```
            lineWidth: 10,
            circularStrokeCap: CircularStrokeCap.round,
            center: Text(d2),
          ),
        ]
      )
    ),
```

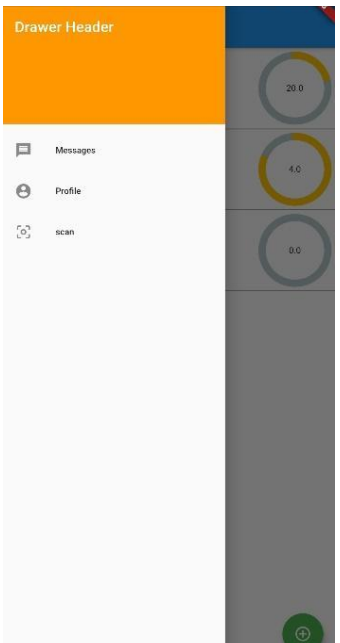
Fig 6 : Datapage Design

The container is an outline widget that surrounds the main widget. We can provide specifications to the container by tweaking its property values. Inside the container we will have a second contaioner that will contain the label of the quantity under measure.

Outside that container we will have the circular progress indicator method to keep track of the values and proportions. To position these widgets properly we can provide edge insets that will allow us to accurately position all the widgets properly with respect to the screen. These methods can be repeated multiple times as per the required amount to display all the data over the application.

Next we need a sliding drawer that will allow us to navigate to and fro to the other pages freely. To create this we need to put up a simple method drawer:

```
Drawer(
      child: ListView(
        padding: EdgeInsets.zero,
        children:  <Widget>[
          DrawerHeader(
            decoration: BoxDecoration(
              color: Colors.orange,
            ),
            child: Text(
              'Drawer Header',
              style: TextStyle(
                color: Colors.white,
                fontSize: 24,
              ),
            ),
          ),
          ListTile(
            leading: Icon(Icons.message),
```



```
title: Text('Messages'),
),
ListTile(
  leading: Icon(Icons.account_circle),
  title: Text('Profile'),
  onTap: () {
    Navigator.push(
        context,
        MaterialPageRoute(
          builder:(context) => ProfileScreen(),
      )
    ); // close the drawer
  },
),
ListTile(
  leading: Icon(Icons.center_focus_weak),
  title: Text('scan'),
  onTap: () {
    Navigator.push(context,
        MaterialPageRoute(
          builder: (context) => ScanScreen(),
        )
```

Fig 7 : Drawer Design

This method will create a drawer and will also create buttons with which we can navigate to the other pages of the application.

2.3.9 Recording and Processing Data from Firebase Database

Once the Data page is ready we need it to be able to draw store data from the Firebase database and correct it format into a more useable local format. To begin reading and streaming of data from the firebase database we will need to use the following method.

```
final dataDist = FirebaseDatabase.instance.reference();
```

This method will link us to a current latest version of the firebase real-time database and allow us to stream and record data. Next we will create a methods that when called takes a snapshot of the data on the firebase database and provides us with the appropriate values as required.

```
dataDist.once().then((DataSnapshot dataSnapshot){
    Map<String,dynamic> myMap = new Map<String,dynamic>.from(dataSnapshot.value);
```

This snapshot then needs to be broken down and then converted into strings and double formats thereby providing us with appropriate values that can be applied by the circular percent indicators.

```
double data3 = myMap['distance3']/100;
    double data1 = myMap['distance1']/100;
    double data2 = myMap['distance2']/5;
```

Once we convert the data into double values it becomes much easier to manage the data and display it as the circular percent indicator requires data inputs to be in the form of percentage of the original value we convert the data as required

```
d1 = (lith*100).toString();
    d2 = (data2*5).toString();
    d3 = (neo*100).toString();
```

After we convert the data we can also display it by converting the percentage data back into strings from double value and allows us to the use this data for further applications.

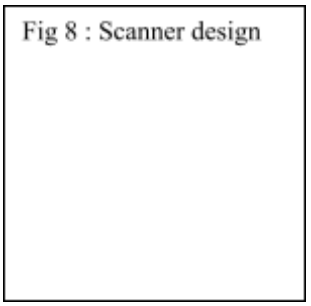
2.3.10 Creating a Scanner

To add data to the list view on the data page we have the firebase database but to manually add items to the list we can have an additional methods that involves scanning tagged items that are induced with specific QR or barcode and allowing them to be added to the list view.

To do so we need to create a scanning method that will use the devices camera as a scanner. After we position the scanner page and we give it a basic layout we can code in the scanner by using the following method.

```
Future scan() async {
  try {
    String barcode = await BarcodeScanner.scan();
    setState(() => this.barcode = barcode);
  } on PlatformException catch (e) {
    if (e.code == BarcodeScanner.CameraAccessDenied) {
      setState(() {
        this.barcode = 'User did not grant the camera permission!';
      });
    } else {
```

```
      setState(() => this.barcode = 'Unknown error: $e');
    }
  } on FormatException{
    setState(() => this.barcode = 'user returned without input');
  } catch (e) {
    setState(() => this.barcode = 'Unknown error: $e');
  }
  return barcode;
}
```



After this we link all the pages together using the navigator command lines and the application will be ready for debugging and compiling. We use the following commands from the navigator command set.

Navigator.push(context, MaterialPageRoute(builder: (context) => ScanScreen()));

We can modify the above command line as per our requirement and create multiple page routes.

CHAPTER 4
RESULT ANALYSIS

The project mainly focuses on collecting of information from sensors and real time transferring of these information to the primary user, the project mainly focused on calculation of the precise information from the object under observation and by using Arduino Yun and the Firebase webserver updating of the said information to the user's phone. For the current experiment only one kind of the sensors were used, but multiple sensors of the same kind were used to calculate different things.

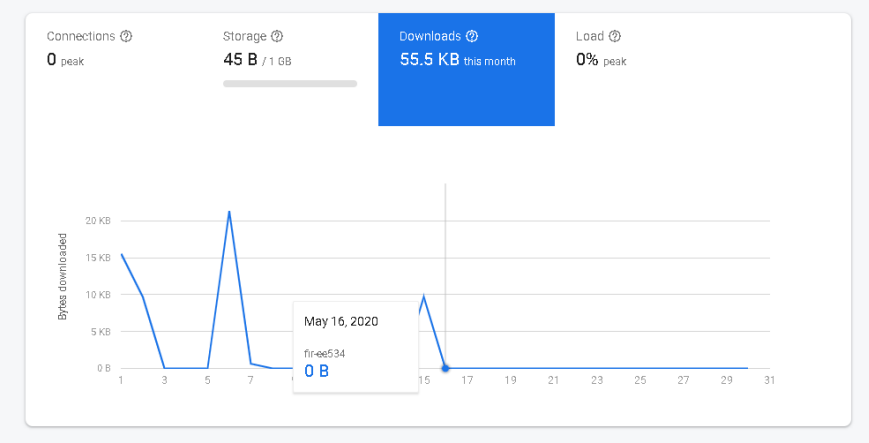
As represented above the different outputs can be observed, this is done by dumping the required code in the Arduino and the measurement is done in a periodic manner or it can set to work when the user sends the request signal, on receiving the signal the Arduino runs the required code and proceeds to send the result via the Firebase server to the android.

In case of the mobile software phase of the project, the main focus was to download the data, which was uploaded onto the firebase database server, and to properly integrate it onto a safe and secure application. To make the application more secure the there was also a requirement of the implementation of a secure login method.

CHAPTER 5
CONCLUSION AND FUTURE SCOPE OF WORK

The objective of this project was to create a hardware which was attentively able to gather information form a container containing the object under testing and send that information to wirelessly to the user. In this case an Arduino Yun was used as the host which was able to gather the information from the sensors and by using the Wi-Fi was able to send the information to an online server in real time so that the information can be accessed by the user any time and from anywhere as long as both the Arduino and the user's device has the internet connection.

So in general the System thus created was able to gather the information in real time and wirelessly transfer the said information of the object under testing to the user's device. Thus proving instant access to the said information.



So from the above graphs and columns we can observe the information that is sent to the Firebase server is sent to the android with precision.

For future application, this project can be implemented in several fields and professions, from a simple household to tracking the stocks of an entire business and improve the capacity of the profit production and comfort.

This project is fully scalable and as these solutions are virtual, upgrades are doable at any time. And depending on the user the experience can be modified to make the experience more satisfactory and more comfortable and efficient. This will also potentially reduce the amount of paper documentation required, to keep stocks and details, potentially to zero with the right application. Due to this the business/projects can have the budget shifted elsewhere to improve their services further.

The use of an online database also allows us to record the older statistics and predict the future. Not having precise data on these metrics can negatively influence your business. If you're not making accurate predictions, you're susceptible to wasting money on surplus stock when demand is low. You're also at risk of missing opportunities to grow if your product sells out sooner than you anticipate.

With cloud-based inventory management, current reports help you make better decisions about the future of your products. Not only are you able to predict proper stock levels, but you can also identify the best times to switch up your products, redevelop your marketing strategies and introduce new services.

REFERENCES

[1] Madhuram. M, Ashu Kumar and Pandyamanian. M., "Cross Platform Development using Flutter" *International Journal of Engineering Science and Computing,* April 2019.

[2] Nilanjan Chatterjee, Souvik Chakraborty, Aakash Decosta and Dr. Asoke Nath " Real-time Communication Application Based on Android Using Google Firebase " *International Journal of Advance Research in Computer Science and Management Studies, Volume 6, Issue 4, April 2018*

[3] Jenifer and D. John Aravindhar., "Iot Based Air Pollution Monitoring System Using Esp8266-12 With Google Firebase," *2019 J. Phys.: Conf. Ser. 1362 012072 doi:10.1088/1742-6596/1362/1/012072*

[4] Chunnu Khawas and Pritam Shah" Application of Firebase in Android App Development-A Study" *International Journal of Computer Applications (0975 – 8887) Volume 179 – No.46, June 2018*.

[5] DIVYESH ZANZMERIYA and ANKITA PANARA ., "Implementation of Industrial Automation Systems using Raspberry pi by IOT with FIREBASE" *International Research Journal of Engineering and TechnologyVolume: 05 Issue: 05 | May-2018*

[6] Robert Sjodin and Mohamed Lotfy., "Developing cloud-based mobile apps using google firebase," *2019 Journal of Computing Sciences in Colleges October 2019*

[7] Abbas Mohammed Younus Yousif., "Controlling for IoT Projects Systems based on Smart Phone using Firebase Notification Technique," *International Journal of Computer Applications, Vols 181 number 33,* 2018

[8] Shweta Singh, , Shikha Verma, Surendra Kumar., "Home Automation Using Node MCU, Firebase & IOT," *International Journal of Scientific Research and Review Volume 07, Issue 03, March 2019*.

[9] Leo Louis, “Working Principle of Arduino And Using It as a Tool For Study and Research” International Journal of Control, Automation, Communication and Systems (IJCACS), Vol.1, No.2, April 2016

[10] V A Zhmud, N O Kondratiev, K A Kuznetsov, V G Trubin, L V Dimitrov, “Application of ultrasonic sensor for measuring distance in robotics”, IOP Conf. Series: Journal of Physics: Conf. Series 1015 (2018) 032189

[11] Zaw lin Oo, Theit Win Lai, Su Mon Ko, Aung Moe. “IoT based Weather Monitoring System Using Firebase Real Time Database with Mobile Application”, International Symposium on Environmental-Life Science and Nanoscales Technology 2019 (ISENT2019).

[12] D. Bruneo, S. Distefano, F. Longo, G. Merlino, A. Puliafito and A. Zaia, "Head in a Cloud: An approach for Arduino YUN virtualization," 2017 Global Internet of Things Summit (GIoTS), Geneva, 2017, pp. 1-6, doi: 10.1109/GIOTS.2017.8016263

[13] forum.arduino.cc

[14] https://developer.android.com/docs

[15] Madusanka, Samarappulige. (2016). Busy programmer’s guide to Firebase with Android..

[16] http://developerfirebase/android.com.

[17] http://.en.wikipedia.org/wiki/Internet_of_things

[18] http://firebase.google.com/products/realtime-database

[19] http://flutter.dev/docs

[20] console.firebase.google.com

PROJECT DETAILS

Student Details			
Student Name	**Neil N. Kadam**		
Register Number	160921202	Section / Roll No	A/34
Email Address	Emailneil1998@gmail.com	Phone No (M)	7506134152
Student Name	**Abhishek N.**		
Register Number	160921298	Section / Roll No	A/
Email Address		Phone No (M)	
Project Details			
Project Title			
Project Duration		Date of reporting	
Expected date of completion of project			
Organization Details			
Organization Name			
Full postal address with pin code			
Website address			
Supervisor Details			
Supervisor Name			
Designation			
Full contact address with pin code			
Email address		Phone No (M)	
Internal Guide Details			
Faculty Name			
Full contact address with pin code	Dept. of Instrumentation & Control Engineering, Manipal Institute of Technology, Manipal – 576 104 (Karnataka State), INDIA		
Email address			
Co- Guide Details(if any)			
Faculty Name			
Full contact address with pin code	Dept. of Instrumentation & Control Engineering, Manipal Institute of Technology, Manipal – 576 104 (Karnataka State), INDIA		
Email address			

General Guidelines (Delete this page when making the report submission)

- ✔ Project Report to be minimum 40 pages. Reports less than 40 pages will be rejected. Not more than 60 pages.

- ✔ Paper Size: A4; Left = Right = Top = Bottom Margins = 1”
- ✔ Page Numbering Position: Bottom with right justified and continuous numbering from the Introduction Chapter
- ✔ Use Times New Roman Font with Normal Style, paragraph justified and 1.25 line spacing

- ✔ Paragraph Heading: Times New Roman Font, Bold, Font Size 14; Paragraph Matter: Times New Roman Font, Normal, Font Size 12;
- ✔ Sub-paragraphs be appropriately numbered as in 1.1, 1.2, 1.3 etc.; Sub-paragraph Heading: Times New Roman Font, Italics, Font Size 12; Sub-paragraph Matter: Times New Roman Font, Normal, Font Size 12;

- ✔ Figure captions below Figure with chapter wise numbering
- ✔ Tables captions above Table with chapter wise numbering

- ✔ All references must be quoted in ascending order (follow IEEE format for referencing)
- ✔ Project Details page must be the last page in the project report
- ✔ Only Soft bound reports will be accepted. The outer cover of the report will be **PURE WHITE** with a 170 GSM paper.

- ✔ **Arrangement of contents**

  [1] Cover page (same as inner page)
  [2] Inner page

[3] Dedication (Optional)
[4] Certificate
[5] Certificate on company letter head
[6] Acknowledgement
[7] Abstract
[8] List of Tables
[9] List of Figures
[10] Table of contents
[11] Chapters 1, 2, 3, 4, 5
[12] References (follow IEEE format)
[13] Annexures
[14] Project Details (Last page of the report)

- ✔ The above guidelines should be used only as a help guide and is more or less a standard way of report writing
- ✔ Report formatting should not be disturbed in any form
- ✔ **Project students are requested to <u>discuss with their department guides</u> regarding the contents of the project report**
- ✔ Hard Copies to be prepared: Will be intimated.
- ✔ Soft copy (both word and pdf format) to be submitted in Department LAB with project title as the file name. .