CUSTOMISABLE CLOUD-BASED IOT MONITORING SYSTEM WITH SMARTPHONE INTERFACE

BY

ANG CHENG KEE

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2020

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface

Academic Session: Jan 2020

I ANG CHENG KEE

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

[illegible]	[illegible]
(Author’s signature)	(Supervisor’s signature)
Address:	
113, Jalan Keris 16,	
Taman Puteri Wangsa,	Dr. Cheng Wai Khuen
81800 Ulu Tiram, Johor.	Supervisor’s name
Date: 19 April 2020	**Date**: 23 April 2020

CUSTOMISABLE CLOUD-BASED IOT MONITORING SYSTEM WITH SMARTPHONE INTERFACE

BY

ANG CHENG KEE

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2020

DECLARATION OF ORIGINALITY

I declare that this report entitled "**CUSTOMISABLE CLOUD-BASED IOT MONITORING SYSTEM WITH SMARTPHONE INTERFACE**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature	:	[illegible]
Name	:	Ang Cheng Kee
Date	:	19 April 2020

ACKNOWLEGDEMENTS

I would like to express thanks and appreciation to my supervisor, Dr. Cheng Wai Khuen and my moderator, Dr. Alex Ooi Boon Yaik who have given me a golden opportunity to involve in the Internet of Things field study. Besides that, they have given me a lot of guidance in order to complete this project. When I was facing problems in this project, the advice from them always assists me in overcoming the problems. Again, a million thanks to my supervisor and moderator.

Other than that, I would like to thank my project teammate, Chia Shun Cheng who has provided a lot of assistance to me when completing this project. Although both of us are having different project and task scope, he is still willing to support me when I faced difficulties in developing this project.

ABSTRACT

This project is regarding a growing trend – Internet of Things (IoT). IoT is a collection of multiple things that connected to the Internet and performing data collection, sharing, and processing. As the IoT trend is growing at an incredible speed, people begin to concern about the challenges raised from IoT. For example, security, user-friendliness, privacy, and customizable issues. There are two major problems will be the focus on this project – customisation and user-friendliness. As most of the existing IoT applications are unable to fulfil a user-friendly and customisable IoT monitoring system, there is still a huge learning gap for IoT users in terms of understanding and customising their IoT solution. Some researches and literature reviews are carried out to look deeper into IoT customisation and user-friendliness issues. The reviews of the products including IoT in a Box, EzBlock Studio, RaspController, IoTool, Lego Mindstorm EV3, SimpleLink SensorTag, Xiaomi Home and Skydrop Smart Sprinkler are written in the literature review section of the report. After reviewing the strengths and weaknesses of existing products, a clearer vision regarding the IoT challenges will be presented in the report. The solution of this project is to develop a customisable cloud-based IoT monitoring system with smartphone interface in order to assist users in learning IoT and customising preferable IoT solutions easily. The final product of the project is the combination of IoT hardware, Google Cloud Platform, Firebase and an Android mobile application with the ability to connect, control, monitor and customise users' IoT solutions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

5G	Fifth Generation
API	Application Programming Interface
CPU	Central Processing Unit
GPIO	General Purpose Input/Output
IOT	Internet of Things
IP	Internet Protocol
MQTT	Message Queueing Telemetry Transport
RAM	Random Memory Access

CHAPTER 1: INTRODUCTION

1.1 Problem statement

According to recent research from Business Insider, IoT Analytics, Gartner and Intel, there will be 64 billion IoT devices connected in the world by 2025 and investment is gearing up in IoT solution with 5G technology on the horizon. (Petrov, 2019). The increasing number of IoT devices in recent years indicates that there will be a huge number of users using IoT solutions as well. However, there are some existing problems of IoT solutions in the market which are customisable and user-friendliness issues that cause a gap between end-user and IoT solutions.

In the ecosystem of IoT, the users always concern about the customisable of IoT solutions. They are always demanding an IoT solution to perfectly fit their requirements and solve the problem facing as perfectly as possible. (Kamat, 2017). For example, a farmer needs a specific IoT solution that contains soil moisture sensors, humidity sensors and other specialised sensors and actuators needed in the agricultural industry in order to monitor their farms and crop conditions. However, an aquaculture industry owner will need an IoT solution that includes the monitoring of water conditions to fulfil the industry requirement. Even though in the same industry, different users may have different requirements for IoT solutions to solve their problems. For instance, a person may want to include a light sensor and a temperature sensor in his smart home solution but another person may not. Hence, the market demand drives to the growth of different IoT providers to develop specific IoT solutions for different kinds of users. When the users need an IoT solution, they always need to find a provider to develop a specific IoT solution.

The problems come when people wish to customise the solution in the future. It means that the users need to customise their existing solution or purchase a new IoT solution which is not cost-effective if the degree of changes required is small. From the perspective of IoT solution users, customisation of IoT solution means that integrate different IoT components to a solution according to users' preferences. (Pande, 2017). When talking about customisation of IoT solutions, it is crucial for the users to work with reliable IoT providers. (Krupitzer, n.d.). It requires a close interaction between the users and the IoT solutions providers. The users need to work closely with the providers through the product design phase and manufacturing phase in order to produce a solution that fits their needs. (Yang et al., 2017). It may be not an issue when the users and providers are always free to interact with each other, the IoT solutions is easy to be changed, and the requirements of the solutions are clear enough from the users. But it comes to a big problem when the user or provider unable to closely interact with each other, or when the existing solution is too expensive or technically difficult to be changed, and the worst case is the ambiguous requirement of the users for the new IoT solution. Sometimes, users will prefer to customise their IoT solution by themselves but it is a tedious process for those non-tech-savvy users as there is a learning gap between the existing IoT platform to customise their solutions. Figure 1.4 shows the trade-off for a different type of IoT solution. If users tend to buy a standardized IoT package, cost efficiency and flexibility may become a problem in the future. If the users tend to develop an IoT solution fully from scratch, implementation speed may become a concern as well. The best practice should be building a customisable IoT platform to support the solution. (Hackbarth, 2016).

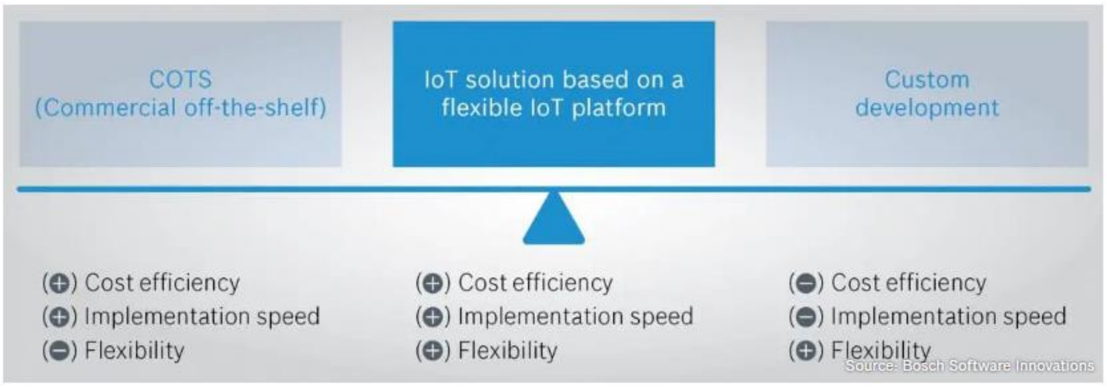


Figure 1.1 Infographic from Bosch Blog

In short, although there are some existing customisable IoT platforms in the market, there is still lacking of a customisable cloud-based IoT platform for the users to customise their IoT solution in terms of data visualisation, monitoring, multiple 'things' support and 'things' automation.

Other than customisable, user-friendliness is another biggest concern of the users in the IoT environment. No matter how powerful is a product, when nobody knows how to use it, it is meaningless to have the product. The concept is the same when talking about the user-friendliness of an IoT solution. An IoT app interface should be as friendly as possible as most of the people is not willing to read through a full user manual before using an IoT solution. (Eliftech, 2019). Although there is a huge number of IoT solutions in the market, there is still a big learning gap between users and IoT apps. Figure 1.3 is one of the infographics produced by Metova survey in 2018 which shows that 80% of IoT users do not even understand the meaning of IoT. The user-friendliness problem occurs when users tend to customise their own solutions. One of the challenges in terms of user-friendliness is the difficulty of onboarding process for IoT solutions. For example, users are required to have some technical knowledge regarding IoT environment such as IP address, Port SSH and other related information in order to set up a connection to the application provided. Besides, users are required to have certain programming knowledge in writing scripts to monitor and control their IoT solutions. Last but not least, some existing applications require users to write code in order to allow different IoT devices in order to communicate with each other. Again, the users may face difficulty if they do not have any knowledge regarding programming language. **Although the concept of user-friendliness is vital in the development of IoT interface, there are still a lot of IoT interfaces not able to achieve a user-friendly design to the user in order to reduce the learning gap of customising an IoT solution.**

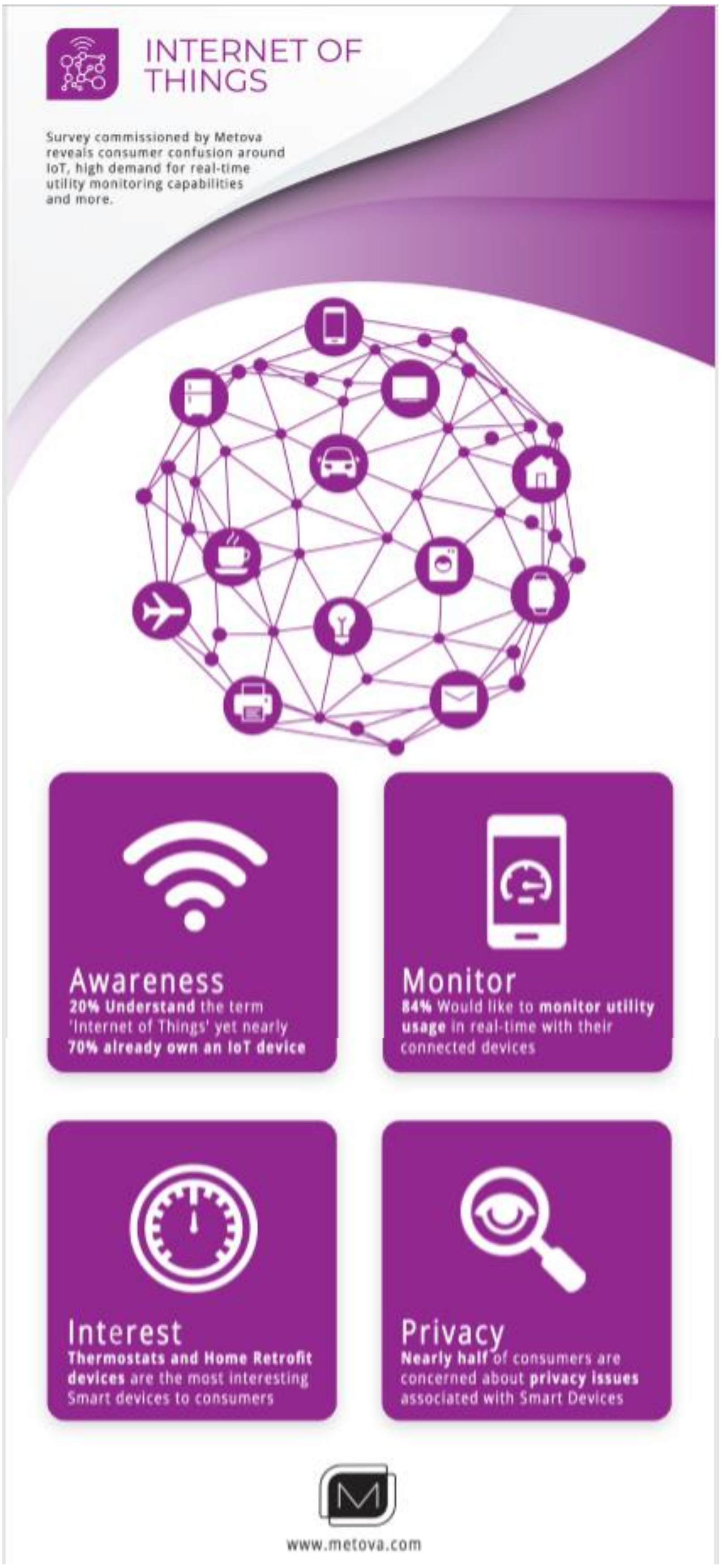

INTERNET OF THINGS
Survey commissioned by Metova reveals consumer confusion around IoT, high demand for real-time utility monitoring capabilities and more.
Awareness
20% Understand the term 'Internet of Things' yet nearly 70% already own an IoT device
Monitor
84% Would like to monitor utility usage in real-time with their connected devices
Interest
Thermostats and Home Retrofit devices are the most interesting Smart devices to consumers
Privacy
Nearly half of consumers are concerned about privacy issues associated with Smart Devices
www.metova.com

Figure 1.2 Infographic from Metova Survey

1.2 Background Information and Motivation

Nowadays, people always talk about the term “Smart”. For instance, smart home, smart agriculture, smart car, or even a smart city. All of these smart things are formed from the innovation of the Internet of Things (IoT). First of all, what is the Internet of things? Internet of things is a collection of multiple devices that are connected to the internet and performing data collection and data sharing. (Ranger, 2018). Almost every device can be connected to the Internet such as sensors, lamps, fans, manufacturing machines, washing machines and so on. For example, all of the lamps and fans in a house can be connected to the cloud and this can be known as a smart home. Few of the farms can be connected together via the Internet and form smart agriculture. For a broader scale, all of the traffic lights in a city can send data to the cloud and form a smart city. Normally, the IoT devices are connected to the IoT gateway or another edge IoT device to analyse the data via the cloud or locally. (Rouse, n.d.). These IoT devices can work together by sharing their data, act on to the data collected and even communicate with each other. In fact, IoT was just a concept before the year 1999. This concept was only put in practice in the year 2013 and evolved to a system by implementing multiple technologies that ranged from micro-electromechanical system to the embedded system and from the Internet to wireless connection. (Foote, 2016).

According to analyst firm Gartner, they estimate that there will be over 26 billion connected devices by 2020. (Morgan, 2014). This indicates that the trend of IoT is growing rapidly and the estimation by analyst firm Gartner can be verified now according to the recent Juniper research. According to the research, the current projected number of connected IoT devices is almost 21 billion and the number will continue to grow until 50 billion in 2022 which is 140% of the growth of the IoT devices in the world. (McKendrick, 2018). The speedy growth of IoT shows that the increasing demand for IoT and it is really bringing huge impacts to our world. Figure 1.1 and figure 1.2 are some of the infographics regarding IoT from the CPA Canada Profession in 2019. From the infographics, the IoT is growing at an incredible speed and the spending will surpass 1 trillion in 2022. (Lajartre, 2019).

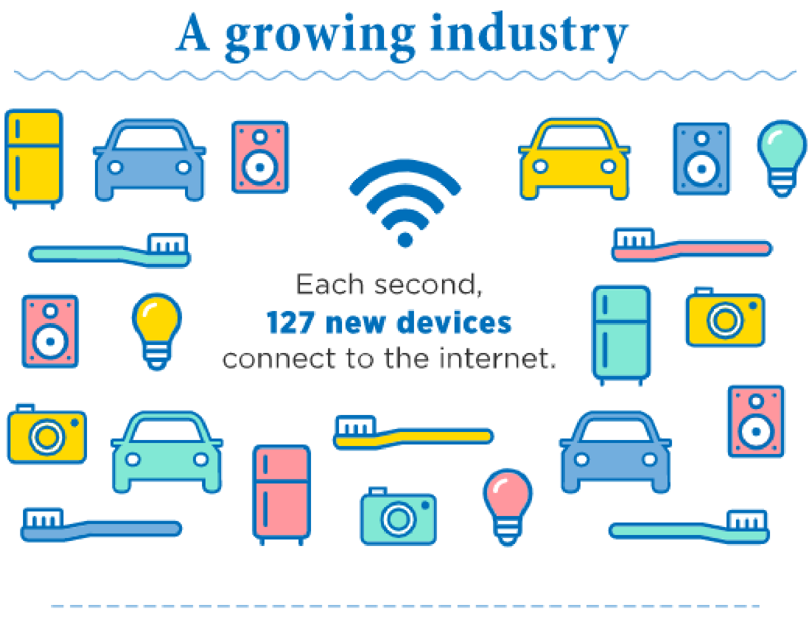


Figure 1.3 Infographic from CPA Canada Profession

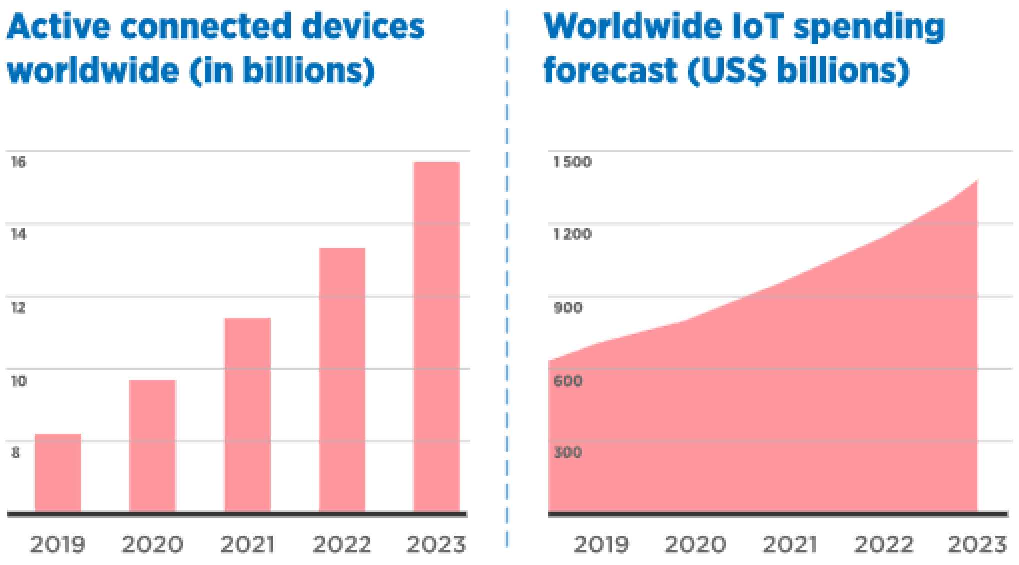


Figure 1.4 Infographic from CPA Canada Profession

The way of people doing things is changing due to the implementation of IoT. For example, some people nowadays fully implement the technology of IoT in their homes such as controlling the lamps using voice or automatic watering in the house's garden. A lot of existing IoT products such as Amazon Alexa, Google Assistant, Samsung SmartThings Hub are some of the most popular smart home ecosystems that available today. They help to control your home devices, monitoring the environmental conditions, analysing the data collected, and perform other features according to user preferences. By implementing the smart home concept with IoT devices, house environment and energy usage can be monitored in a more efficient way and also enhancing residents' social well-being. (Bedi, Kumar Venayagamoorthy and Singh, 2016)

IoT is not only changing the meanings and experience of going home, but it also emerges into the industry level as well. After the innovation of IoT, people start to look at the integration of IoT technology into different industry areas. The merging of IoT into the industrial manufacturing system has changed the trend of the manufacturing paradigm. (Xu, Xu and Li, 2018). According to IDC statistics in 2017, the manufacturing industry has spent about $102.5 billion in the year 2016 for IoT technology implementation in the industry. (i-SCOOP, n.d.). This indicates that the trend of IoT is growing rapidly as well in industrial areas in the age of Industry 4.0. Before the IoT technology being introduced, the manufacturing industry is performing its business in a traditional manner which means the products are simply manufactured and delivered to the customer. It may lead to some difficulties in providing customer service, managing and tracking the sold products. With the implementation of IoT in the manufacturing industry, the products can be smarter, trackable, and manageable. For instance, Hilti is a company delivers products, systems and software to the construction industry to make the construction work easier. With roughly 250,000 daily customer interaction, Hilti always enforces on customers' need by innovating new ideas in their business process. Hilti company equip their company products with sensors and connectivity to utilise the construction tools. (Hilti Corporation, n.d.). With the data collected from the physical devices such as power tools, the condition of the tools can be analysed easily to improve the efficiency of product usage, ease the products' maintenance, and also build a better company-user relationship. Figure 1.3 is an infographic from the Hilti company about the implementation of smart tags in their products.

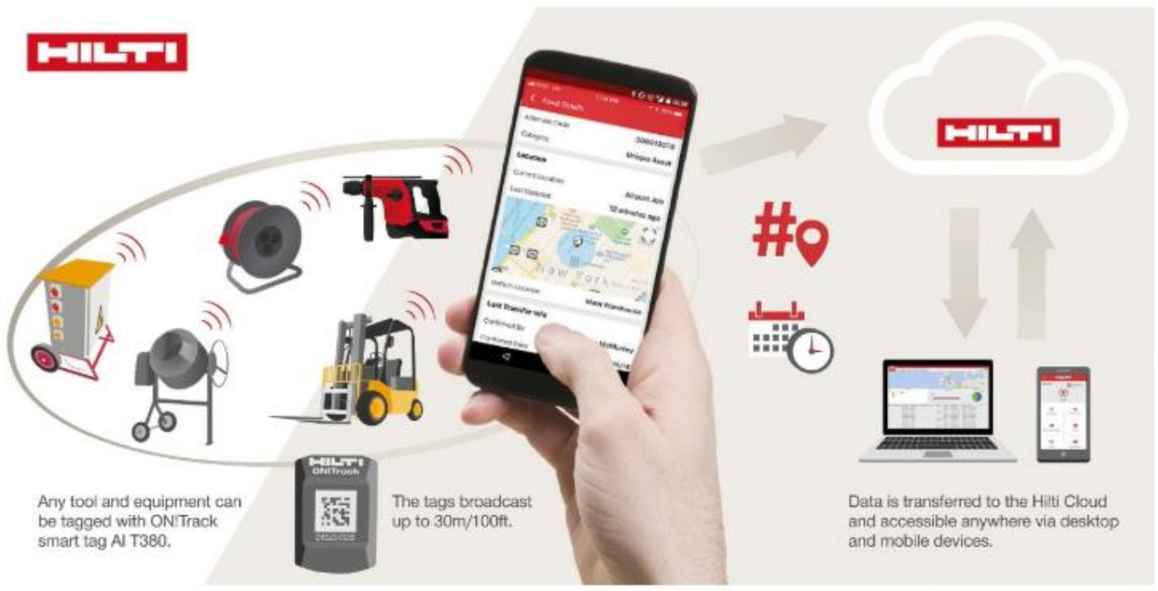

HILTI
Any tool and equipment can be tagged with ON!Track smart tag AI T380.
The tags broadcast up to 30m/100ft.
Data is transferred to the Hilti Cloud and accessible anywhere via desktop and mobile devices.

Figure 1.5 Infographic from Hilti Corporation

Other than the manufacturing industry, IoT technology is also playing a critical role in transforming the agriculture industry in a smarter way. Due to the increasing demand for food, the growth of the agriculture industry has been boosted up to fulfil the needs of the market. Smart farming is introduced now and the way of farming is no longer similar to the traditional farming method. In order to increase crop productivity in terms of quality and quantity, high-tech technologies such as IoT and cloud computing are implemented in the agriculture industry. The IoT solutions will play as the important components to modernise the way of farming whereas cloud computing will store a large amount of data collected from the IoT devices, and perform data analysis to automate the farming process, predict yield, monitoring farm environment and other useful features through a cloud platform. (Tzounis et al., 2017). For example, when monitoring the farm environment conditions such as temperature, humidity, and soil moisture with sensors connected to the cloud, some of the farming processes can be automated as well such as automated watering based on soil moisture and humidity level. By making use of field monitoring and automation of farming processes using IoT, traditional agricultural problems can be overcome such as collecting information throughout the whole farming cycle, performing analysis and prediction to aid the farmer in better decision making. (Mohanraj, Ashokumar and Naren, 2016).

In short summary, IoT technology is one of the major development trends in the current ecosystem. For instance, there is a lot of IoT platform developed to support IoT solutions such as Google platform, Amazon platform and Microsoft platform. IoT brings a lot of positive impacts on society, industry, economy and other aspects of our life. Other than the manufacturing and agriculture industry, there are still a lot of IoT implementation areas such as healthcare and transportation The implementation can be ranged from a simple sensor to a huge amount of IoT devices communicate with each other to perform analysis, monitoring, automation, and others computing features. With IoT in our life, people can have a better living experience, farmers can have a better tool to gain competitive advantages, and other industry areas can obtain the benefits from IoT technology as well.

From the background study and problem statement discussed, IoT is merging into a part of human life but followed by some customisation and user-friendliness issues discussed in problem statement section. Hence, the motivation of this project is to solve the problems by developing a customisable cloud-based IoT platform to enhance the use of IoT technology. By providing a user-friendly interface to support the customisation of IoT solutions, the learning gap between the users and IoT technology can be reduced greatly to encourage implementation of IoT solutions by novice users as well.

1.3 Project Scope

In this project, a user-friendly mobile application with cloud computing features is developed to customise IoT solutions. In the application, users are able to connect to IoT devices with minimal user effort. After connecting to the IoT devices, the devices' setting will automatically be detected by the application and show in its interfaces. From the interfaces, users are able to control the connected IoT devices from the control panel and also monitoring the collected IoT data in a user-friendly design manner. Other than that, users can create customisable triggers for controlling and monitoring the connected things based on simple if-then logic. Due to project time constraints, this project only supports a few common types of sensors and actuators. Besides the mobile application, Google Cloud Platform is chosen as an IoT platform to connect the application and also IoT devices. There are some Google Cloud services are used such as Google Cloud IoT Core, Google Cloud Pub/Sub, Google Cloud Function and Google Cloud Scheduler. Other than that, Firebase services such as Firebase Authentication and Firebase Database are used to provide authentication and storage features for this project.

1.4 Project Objectives

1. Develop a customisable cloud-based IoT platform to enable "things" connection, control, and monitoring.

In this project, Google Cloud platform is acting as a middleware to connect multiple IoT hardware and software application. Cloud computing and IoT technology have become two closely affiliated technologies as one provides another a platform to success. (StoneFly, n.d.). The role of a cloud platform in IoT ecosystem becomes extremely important because the greater generation of IoT data collected. With the cloud computing features provided by the platform, users no need to bother about the storage, data processing, and other data monitoring issue anymore. It is also a cost-effective approach for IoT development because the cloud computing platform allow virtualisation of resources with pay for use model. By combining Google Cloud platform, Firebase services and a mobile application, a customisable cloud-based IoT platform is developed to enable multiple things connection, control, and monitoring. In short, users just need to interact with the extremely user-friendly interfaces provided to have their own solutions easily. Furthermore, communication between the things can be achieved by the simple if-then interface provided in the application.

2. Provide a user-friendly IoT monitoring interface for the proposed platform in order to reduce the learning gap of customising an IoT solution.

As mentioned before, user-friendliness is the major concern in IoT development in order to fully utilise the features of IoT products. As stated in the problem statement, customisation of IoT solutions is a tedious task for IoT users. They need to have the related knowledge with existing IoT cloud platform in order to customise IoT solutions to perfectly fit their needs. For example, users may need to know how to set up a MQTT connection between the IoT device and IoT application interface. Hence, another objective of this project is to develop a friendly interface to support the proposed IoT platform. From this interface, users are able to have a quick and easy setup with their IoT devices. By using the interface provided, users can simply plug and play their IoT devices to develop a best-fit IoT solution by themselves. For example, users can simply customise some IoT devices setting like adding or removing sensors from the microcontroller, then the changes will reflect on the application interfaces. Besides, users can control the IoT devices easily from the interface such as switch on and off an actuator. Furthermore, the interfaces also allow user to derive insight from data collected by visualising the data through chart view. Moreover, the communication between devices can be achieved easily by setting automation rules.

1.5 Impact, Significance and Contribution

People are always aimed for a perfect IoT solution to fulfil their current needs or even future requirement development. Normally, there are few ways to do but it is not effective in terms of cost, time, and effort. For instance, users can develop an IoT solution from scratch but it requires technical knowledge in terms of hardware design, software design, selected Cloud IoT platform and other issues. Other than that, users can find a reliable IoT solution providers to develop a solution to meet their current needs. However, when there is a change required in the solution, it becomes a tedious task again for the users and providers. It is normally time and cost consuming to change a pre-packaged IoT solution in the future. Hence, after the completion of this project, there will be some contributions to the customisation and user-friendliness problems mentioned above in the current IoT ecosystem.

From the development of customisable cloud-based IoT platform, the ultimate outcome of this project is to allow users to have their preferable IoT solution easily with the interface provided to support the IoT platform. With the interface provided, there will be almost zero learning gap for the users to customise their own IoT solution. For instance, users are allowed to have a quick and easy setup with their IoT devices from the mobile application. Then, users can create values immediately from the data collected and bypass complicated onboarding process. Without too much training provided to the users, they can simply plug and play the IoT solutions according to own requirement. The simple plug and play concept in this project allows users to clarify what they want when their initial requirement of an IoT solution is ambiguous. This saves the interaction time between users and IoT solution providers meanwhile able to create a best-fit IoT solution to the users. In addition, it will also assist the users in learning the IoT concept. Last but not least, the intuitive user interface provided will assist users in monitoring their IoT solutions and make a better decision from the data collected.

1.6 Highlights of What Have Been Achieved

There are a few highlights of the achievement of this project to fulfil the project objectives:

1. Google Cloud Platform is used to support the things connection, data collection, and data processing.
2. Firebase Realtime Database is used as a backend server to support the create, retrieve, update and delete operation for the mobile application.
3. An Android Mobile application is developed for the end-users to connect multiple things, view connected sensor data, visualize processed sensor data in graph view, control connected actuator, and configure triggers for multiple things using simple if-then logic.

1.7 Report Organisation

This report is organised into 6 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Design, Chapter 4 System Implementation and Testing, Chapter 5 System Outcome and Discussion, Chapter 6 Conclusion. The first chapter is the introduction of this project which includes problem statement, project background and motivation, project scope, project objectives, project contribution, highlights of project achievements, and report organisation. The second chapter is the literature review carried out on several existing IoT applications in the market to evaluate the strengths and weaknesses of each product. The third chapter is discussing the overall system design of this project. The fourth chapter is regarding the details on how to implement the design of the system. The fifth chapter will be the system outcome and discussion. The last chapter will conclude the project reviews, project contribution, project novelties, and future work of the project.

CHAPTER 2: LITERATURE REVIEW

2.1 EzBlock Studio

2.1.1 Product Review

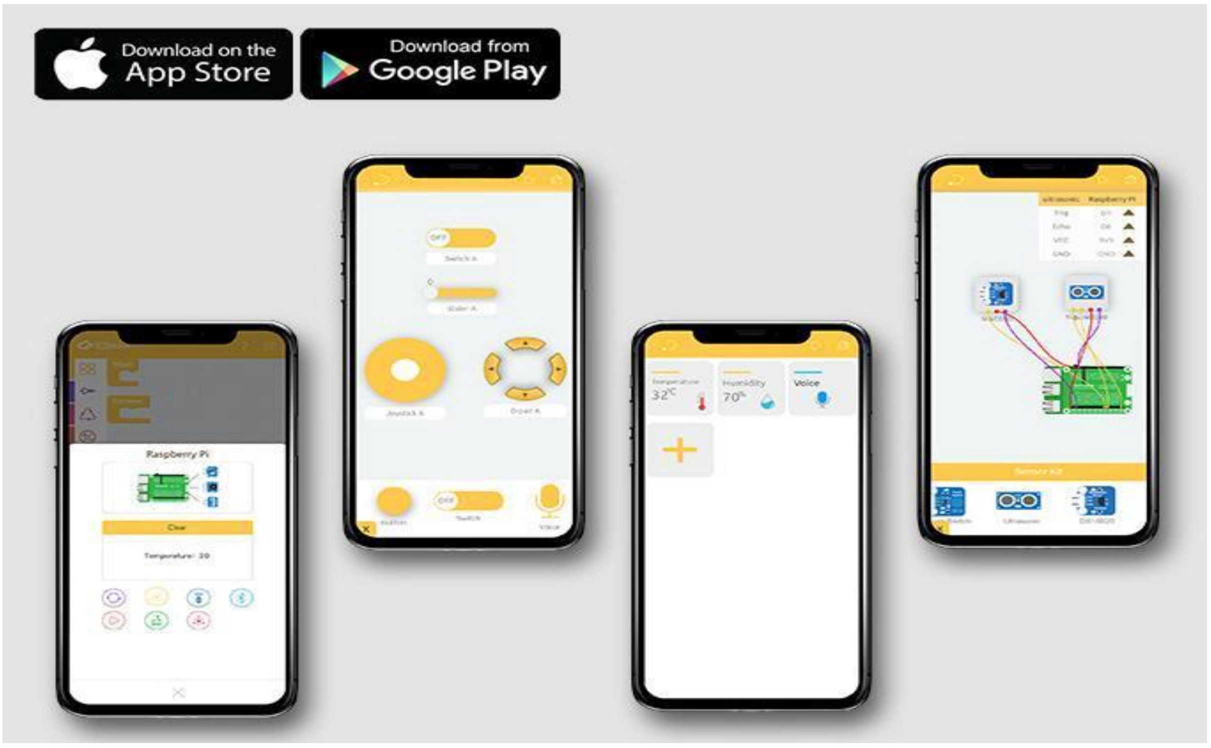


Figure 2.1 Ezblock Studio

Ezblock Studio is a mobile application that works with its hardware products – Ezblock Pi. Ezblock Pi is a controller that can be attached to a Raspberry Pi as an extension to build a connection between Raspberry Pi and Ezblock Studio app. In this IoT product, Ezblock Pi acts as a platform to connect sensors and actuators to Raspberry Pi, whereas the Ezblock Studio acts as the software application interface to monitor the Ezblock Pi. Once the Ezblock Pi attached to a Raspberry Pi, users can start to customise their own IoT solution on the Ezblock Pi and monitor the solution via Ezblock Studio interface. The following parts will explain about features highlight of Ezblock Studio which include virtualisation of hardware circuit, debug and build a prototype of IoT solution, visualisation the data of from sensors, and remote control of IoT device.

First, Ezblock Studio allows users to customise their own IoT solution with block diagram or mobile programming languages such as Python and Switch. By connecting the Ezblock Studio with Ezblock Pi by Bluetooth, users could flash their written code in Ezblock Studio wirelessly to Ezblock Pi in order to set up the configuration of the IoT sensor or actuator attached on Ezblock Pi. Once the connection was set up, users could build up a virtual circuit in Ezblock Studio in order to detect the circuit setup correctness before building the physical circuit. (Ezblock, n.d.). Flash code wirelessly and visualisation of hardware circuits allowed users to build a prototype of their solution quickly and debugging the solution as well via the application interface of Ezblock Studio. Besides, Ezblock Studio also provided an IoT panel for users to remote control and monitoring their IoT solution. In short, Ezblock studio is an application interface to support Ezblock PI with hardware simulation, remote controller, Bluetooth debugger and also quick prototyping of IoT solution. (Kickstarter, n.d.)

2.1.2 Strengths

Ezblock Studio is a powerful tool for IoT users to customise their IoT solutions with the aid of Ezblock Pi. After connecting to the Ezblock Pi, users can build a prototype of their IoT solution using the app by features provided from Ezblock Studio such as flashing code, virtualisation of hardware circuit, and debugging. By supporting the real-time simulation of IoT devices, it makes the building of IoT project become easier in the interface provided. With the control panel provided in this app, users were allowed to remote control their IoT devices when they are not around with the IoT device.

2.1.3 Weaknesses

Although Ezblock Studio was powerful enough to customise an IoT project, but it was not considered as a user-friendly tool for those non-tech-savvy IoT users. For those who had a certain knowledge level regarding IoT hardware and software, it may not be a big challenge for them to set up an IoT project according to their preference. But for those beginners who lack of technological knowledge regarding IoT, building an IoT solution from scratch by using Ezblock Studio may have a huge learning gap to build and customise their project. From a software knowledge perspective, they need to learn about some programming language to set up and configure the sensor and actuator in Ezblock Pi. Besides, they also need to have some knowledge regarding hardware circuits as well in order to set up and visualise it in Ezblock Studio. Hence, this application may be a powerful but meaningless tool for those who want to customise an IoT solution but lack of programming knowledge and hardware knowledge.

2.2 IoT in a Box

2.2.1 Product Review

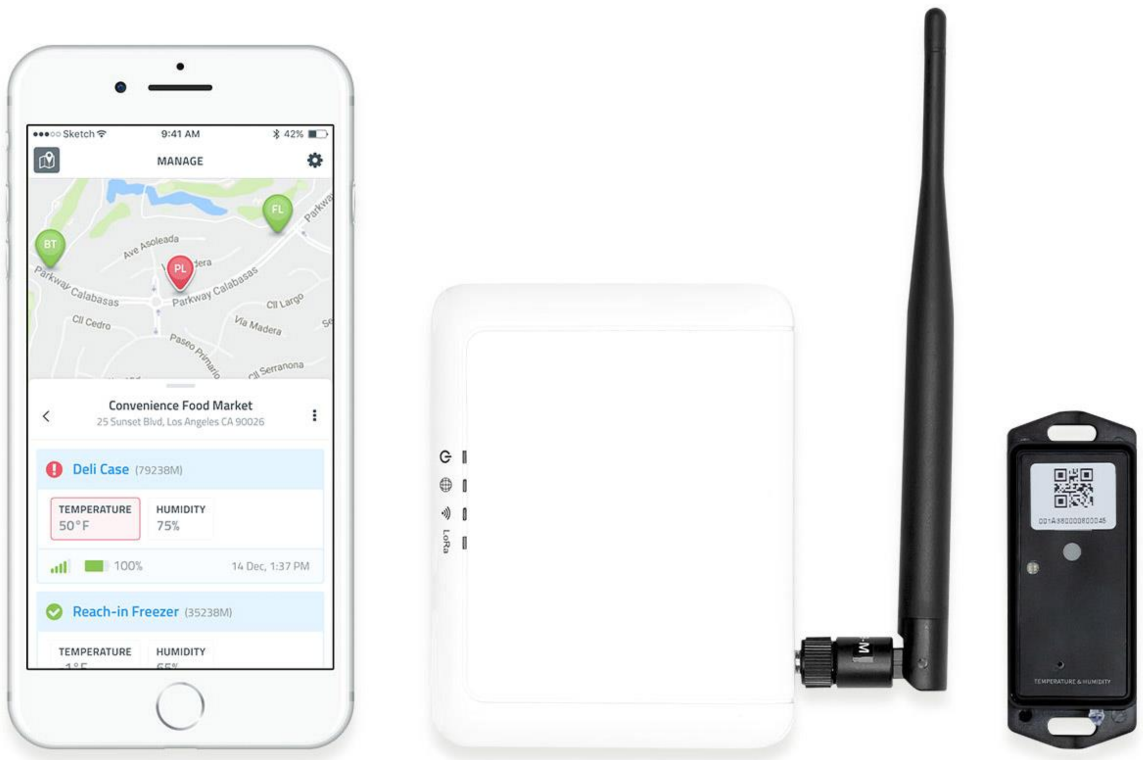


Figure 2.2 IoT in a Box

IoT in a Box is one of the customisable IoT products in the market with the combination of the application interface and hardware. IoT in a Box involves in multiple industry areas such as food service, education, healthcare and retails. (Iotinabox.com, n.d.). Customers can contact to solution provider in order to have a suitable IoT solution which includes sensors, and gateway according to customers' requirements. In order to support the communication between IoT in a Box application and the sensors, all of the sensors and gateways are developed specifically to connect to its application.

After purchasing the IoT in a Box hardware, users can easily set up an IoT solution with a preferable combination of IoT hardware via the IoT in Box software application. There are a total of 3 steps to build an IoT solution using IoT in a box. First, scan the QR code attached on the hardware. Once the users download the IoT in a Box from the app store, they can scan the QR code attached on the hardware to set up a connection with the hardware. After connecting to the hardware, install the hardware at users’ preferable location. Then, start monitoring the solution from the application interface. From the application interface, users are provided with multiple features to monitor their solutions. They can view complete sensor data including history, detailed sensor map to show sensor location and status, and get an instant alert message when a problem detected.

2.2.2 Strengths

User-friendliness can be one of the major strengths of IoT in a Box. The super quick and easy setup of an IoT solution using QR code scanning is really an attractive selling point of this product to those non-tech-savvy users. The simple scan and play feature also ease the customisable process of an IoT solution. For example, users can just simply add on a new sensor by scanning the QR code or removing a connected sensor from the application interface within minutes. There is no heavy technical knowledge required for the users to customise their preferable solutions.

2.2.3 Weaknesses

However, IoT in a Box application does not provide a control panel for the users to remote control their IoT devices. Users can only switch on or off their IoT devices’ from the physical devices’ power button. Besides, IoT in a Box may not cost-effective to all of the users. Although the application is free to download in the' app store, however, it only supports IoT in a Box hardware products. The products such as light sensors, temperature sensors are more expensive than the normal IoT sensors in the market. Normal microcontrollers such as Raspberry Pi and normal sensors in the market are not supported by the application as well.

2.3 Skydrop Smart Sprinkler Controller

2.3.1 Product Review

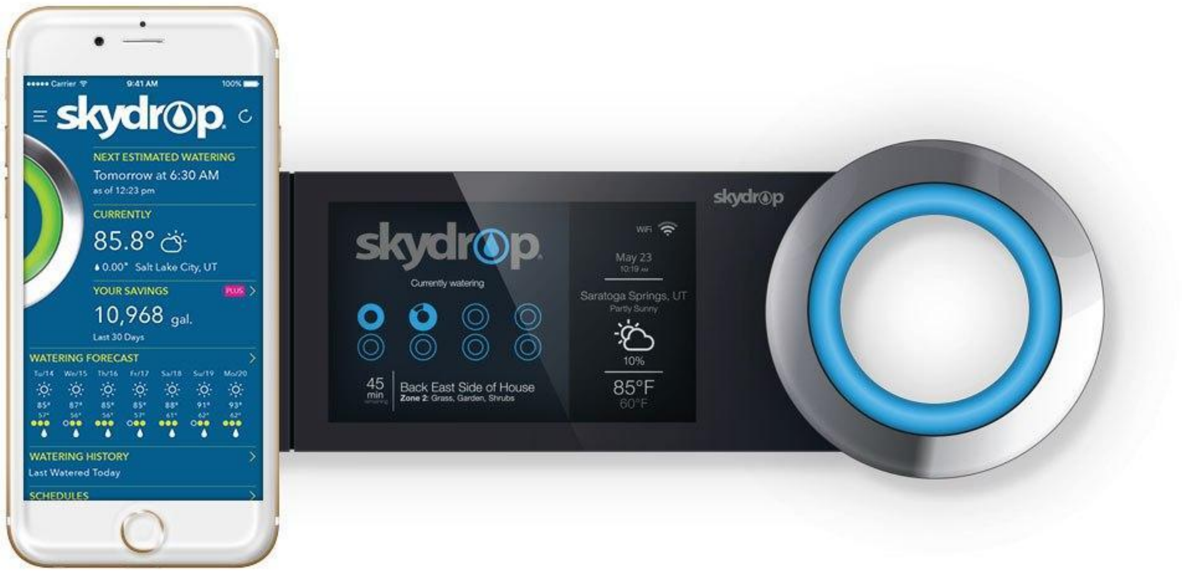


Figure 2.3 Skydrop Smart Sprinkler Controller

Skydrop Smart Sprinkler Controller is a cloud-based interface that adjusts the duration and frequency of the watering cycle according to real-time environmental conditions. (Manhattan Street Capital, n.d.). By connecting the controller to the existing sprinkler system, the Skydrop Controller helps the owner to build a smart watering system in their garden, yard and farm easily. Users can either use the interface of the Skydrop mobile app or the Skydrop Controller to control the sprinkler system. One of the main features is the dynamic watering schedule based on local weather in order to conserve water usage. Some tests have shown that the Skydrop Controller is able to reduce water usage by up to 35%. (Manhattan Street Capital, n.d.). Besides dynamic scheduling, the Skydrop interface also provides a custom schedule based on users' preferences such as soil and plant type. Other than scheduling features, the Skydrop interface also provides other monitoring features such as real-time environmental conditions, push notification when fault detection, multiple control over different watering zones, and watering history report.

2.3.2 Strengths

The most powerful feature provided by the Skydrop Controller is intelligence automation technology. Based on the real-time weather conditions, the Skydrop Controller will automatically update the watering schedule hour by hour to ensure an accurate watering schedule (Skydrop.com, n.d.). Other than that, the Skydrop Controller even forecasts the watering schedule from weather forecasts information collected from weather stations. By using the Skydrop Controller, it almost did every guessing work and scheduling work for the users while maintaining a healthy yard with the most efficiency watering algorithm. Other than that, the Skydrop Controller also integrates with Amazon Alexa service to further enhance users' experiences in controlling their watering settings by using voice control.

2.3.3 Weaknesses

There are some limitations of the Skydrop Controller in terms of customisation of IoT solution. As the Skydrop Controller is a pre-packaged IoT extension to an existing sprinkler system, it may not support multiple types of sensors and actuators from the interface. It means that the future customisation of IoT solutions may not be supported by the Skydrop interface as well. For example, users may not be able to add a more advanced soil moisture sensor to their existing sprinkler system. Besides, setting up Skydrop Controller to the existing sprinkler system still need some technical knowledge of the user regarding wiring, port, and other hardware knowledge. Although the user manual is provided, it is still not considered as a quick and easy installation process for users. Another limitation of the Skydrop Controller mobile application is unable to connect to multiple Skydrop Controller.

2.4 IoTool

2.4.1 Product Review

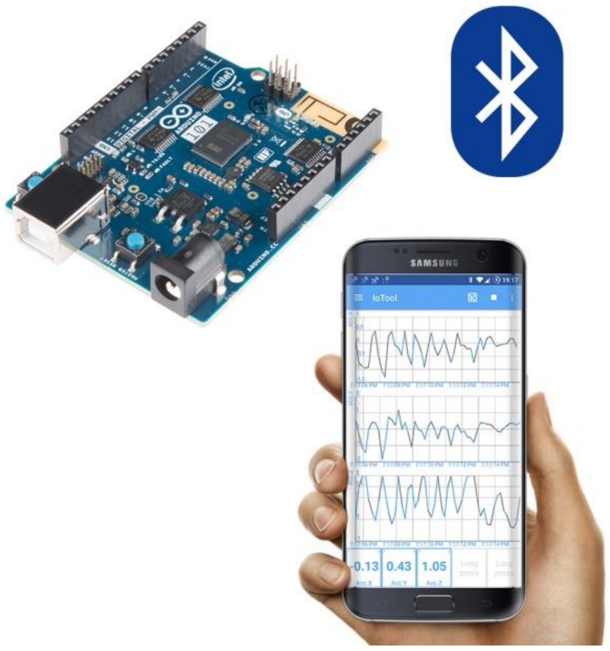


Figure 2.4 IoTool

IoTool is an IoT platform that supports more than 100 types of sensors and 20 types actuators including smartphone sensor, Arduino sensor, Raspberry sensor, cloud-based sensor and even own customised external sensor. (IoTool, n.d.). IoTool consists of many powerful features required in building, monitoring, and customising IoT solutions. First, IoTool provided a dashboard to monitor the real-time or history data collected from the sensor. As IoTool is supporting multiple cloud platforms such as IBM Watson and Amazon AWS, it allows the users to store, access and also synchronise data in the cloud platform. On top of that, by using IoTool, smartphone or Raspberry Pi can become an IoT gateway to connect IoT devices with the cloud platform. (IoTool, n.d.). With the cloud platform supported, users can save their local device storage by storing the data in cloud storage. Other than that, IoTool also provides action trigger function to let the users further control their IoT devices. For example, users can set up a certain condition to trigger the light sensor's action. Furthermore, IoTool also supporting multiple messaging methods such as MQTT, email, and text to ease the communication between devices and users.

2.4.2 Strengths

IoTool performs well in terms of customisation degree of IoT solution. As it is supporting a large number of sensors, actuators and own extensions, it is possible for the users to build most of their preferable IoT solution. Besides, IoT trigger and action function provide users a precise control to IoT devices connected. For example, users can trigger an on and off actions on an actuator by scheduling the trigger.

2.4.3 Weaknesses

As mentioned before, no matter how powerful a product is, if it is too difficult to be used, then the product is useless. IoTool is not considered as a user-friendly application as it is expecting users to read through user manual or having related IoT knowledge using the application. For example, users are required to know how to set up the trigger condition and trigger action together in order to control IoT devices. In terms of user-friendliness, the IoTool is not friendly enough for users to control IoT devices. Instead of providing an intuitive control panel interface, controlling IoT devices in the IoTool interface is still a tedious process by setting up triggers and actions on IoT devices. Other than that, users are required to configure the settings before adding a new sensor. For instance, download an extension for the sensor from the application and read through the sensor manual to set up a connection between the sensor and application. It makes the onboarding process of customizing an IoT solution to become difficult.

2.5 RaspController

2.5.1 Product Review

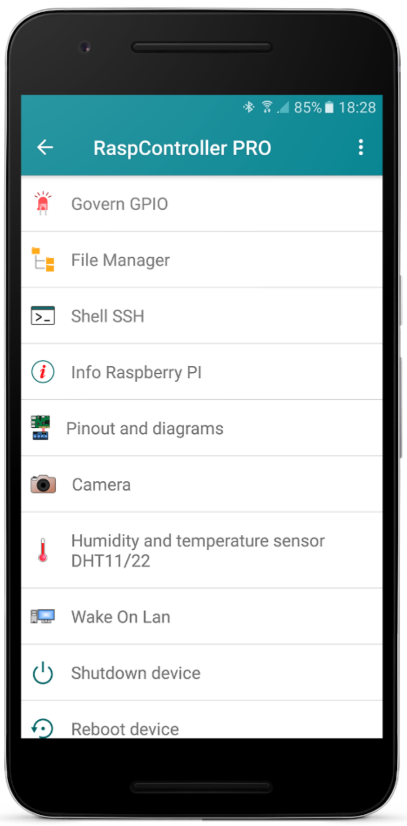


Figure 2.5 RaspController

RaspController is a mobile application developed to serve the purpose of managing Raspberry Pi. By using the interface provided, users can build an IoT solution with a combination of Raspberry Pi, sensors and actuators. It is easy to set up a connection to RaspController once the Raspberry Pi is configured properly. Users need to input the Raspberry Pi setting in the configuration panel such as IP address, password, timeout, port, device name, and username. Then, the wireless connection is set up now if every device's settings are correct. After connecting to a Raspberry Pi, users can perform a lot of features from the application interface to manage the Raspberry Pi. For example, users can control the sensor connected to the Raspberry Pi from GPIO management. Besides, users can also send commands to the Raspberry Pi from the Shell SSH provided in the application. (GallinaEttore.com, 2019). Other than controlling, users can also visualise some sensors data in real-time such as humidity, temperature and pressure from RaspController interface. Besides these features, other features such as File management, Raspberry Pi hardware condition monitoring (Cpu, Ram, Disk) are also provided from this application.

2.5.2 Strengths

RaspController is a great tool for the user to build and customise their IoT solution using Raspberry PI. Once the devices are connected, users can remote control the devices wirelessly with GPIO management and command shell features. It is quite powerful enough for the IoT solution developers who have certain hardware knowledge to play with the Raspberry Pi.

2.5.3 Weaknesses

Although RaspController is powerful enough to develop and customise IoT solutions, the common issue with this kind of powerful controller is the user-friendliness. The RaspController interface is not user-friendly enough and causing a learning gap for those non-tech-savvy users as there is a lot of technical jargon inside the application interface. For example, users must have a certain knowledge level regarding GPIO management and also command in order to control and configure the Raspberry PI. Other than that, the data collected from the sensor may have lesser value when using RaspController as it does not provide any advanced reporting features such as a graph of data history in order to tell the users more stories of the environment. Moreover, the RaspController is only allowed to manage a single Raspberry Pi in a time which will limit the devices' monitoring and controlling features.

2.6 Lego Mindstorm EV3 Programmer

2.6.1 Product Review

Figure 2.6 Lego Mindstorm EV3

Lego Mindstorms EV3 is a customisable robot that brings Lego concept to life by mixing Lego element, motor and sensor (Lego.com, n.d.). The EV3 brick is a powerful programmable computer that allows people to build their own robot by using the motor and sensors. With the powerful EV3 brick, people can build a robot to talk, react, and think with the combination of motor, sensor and programs written. Mindstorms company has developed several applications to support the use of EV3. Lego Mindstorms Programmer is the official programming application developed to support the use of Lego Mindstorms EV3. By using Lego Mindstorm Programmer, users are allowed to build their EV3 robots in the fastest way without the need of any wires or other software. The users can connect the EV3 brick to the tablet app by using Bluetooth. The app provides a drag and drop interface to assists users in building their programming block for the robot. There are some simples programming blocks available such as action block and flow block in the programmer application. Other than the programmer app, Lego Mindstorm commander is another application developed by Lego Mindstorms company to allow users to control their robot remotely without using a computer and wire. By using the Bluetooth connection, users can interact with their robot using their android devices easily.

2.6.2 Strengths

Lego Mindstorm Programmer is an awesome application to allows users to program their robot easily by using intuitive user interfaces. By simply drag the programming block into the interface and click the play button, users can see the result immediately from the robot. The wireless connection between EV3 brick and application also provide an alternative way for the users to connect their robots. Other than that, the Lego Mindstorm Commander is another tool provided to the users in order to control their robot instantly without any wires required. From the user interfaces, users can add their preferable control methods such as slider or joystick to control the EV3 robots. Last but not least, both of the applications also provides some guidelines to guide the users in building their own robots.

2.6.3 Weaknesses

Lego Mindstorm Programmer in tablet or mobile version still lacking of a lot of programming blocks that available in window versions. For instance, if-else logic block and messaging block are not available in the tablet version. This will restrict the users' actions in customising their solution such as communication between the robots. Other than that, lacking of cloud support also become another weakness of the applications. Users are only allowed to remote control or program their robots within the Bluetooth range. The connection between the Lego Mindstorm applications is only one to one relationship which means users are only allowed to control one robot at the same time.

2.7 SimpleLink SensorTag

2.7.1 Product Review

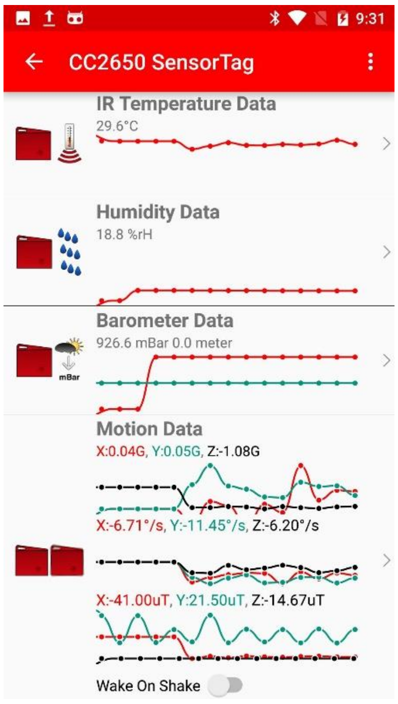


Figure 2.7 SimpleLink SensorTag

SimpleLink SensorTag is an official SensorTag app from Texas Instrument to support Texas Instruments SensorTag Kits. The SensorTag Kits combine the sensor data with cloud connectivity and support multiple types of wireless connections such as Bluetooth low energy, 6LoWPAN and Zigbee. (My.mouser.com, n.d.). There are different types of SensorTag products including wireless MCU Launchpad, CC2650 SensorTag, and others. These products such as CC2650 SensorTag which includes 10 sensors in a small package allows people to connect and visualize the sensor data using SimpleLink SensorTag with no programming knowledge required. (Texas Instruments, n.d.). The CC2650 Sensor Tag is prepackaged with 10 sensors such as temperature, humidity, magnetometer and accelerometer and it is programmable by Devpacks from Texas Instruments. There are 2 ways to connect the SensorTag with the app – wifi or Bluetooth. Both methods require users to turn on the SensorTag and pair it through the app interface. Once the SensorTag is turned on, users can easily pair the tag with the app by Bluetooth or Wifi. By using wifi connectivity, users need to connect the enter the access point settings such as wifi name and password in order to set up cloud connectivity for the SensorTag. When the tag is connected to the application using wifi or Bluetooth, all the sensor data collected will be displayed to the users.

2.7.2 Strengths

SensorTag Kits provide multiple types of wireless connections such as Bluetooth, Wifi, Zigbee and 6LoWPAN. It allows users to customize their IoT solutions with different kinds of connectivity required. SensorTag also allows visualization of the sensor in the cloud platform or mobile app with a graphical view. Other than that, the onboarding process of the SimpleLink Sensor Tag app is considered easy enough for the users to pair the SensorTag with the app by using Bluetooth or wifi connection.

2.7.3 Weaknesses

The difficulty in customize own IoT solutions using SensorTag is one of the obvious weaknesses of SensorTag for novice users. For example, as the CC2650 SensorTag is prepackaged with 10 types of sensors, users may need to customize the tag with their preferable sensors or actuators. However, this process requires users to have some additional hardware and software to customize the solution such as Devpacks and Code Composer Studio. After having the required hardware and software, users still need some programming knowledge to add new sensors or actuators even though some source codes are provided on the official website.

2.8 Xiaomi Home

2.8.1 Product Review

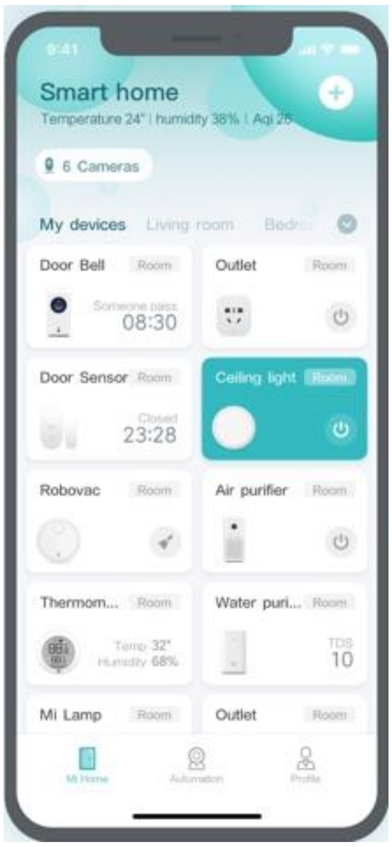


Figure 2.8 Xiaomi Home

Xiaomi enterprise is one of the business Giant in the current market which involving in multiple industries such as the smartphone and IoT industry. The smart home domain is one of the major focuses of Xiaomi in IoT industry. There are a lot of products released by Xiaomi such as sensors and actuators. For instance, temperature sensor, motion sensor, air purifier, Mi lamp and other. These products are easily managed by the Xiaomi Home app which supports centralised control and monitoring of multiple IoT devices. From the Xiaomi Home app, users can easily manage multiple IoT gadgets purchased from Xiaomi by simply adding the gadgets from the interface. From the application, users are allowed to monitor and control their house from anywhere as the gadgets are able to connect to the Internet from Xiaomi Hub or the gadgets itself. For example, temperature and humidity data collected by the temperature and humidity sensor can be updated in real-time to the users through the application. Other than managing those gadgets, automation is one of the features highlighted in the Xiaomi

Home app. Users can set conditions and tasks to customise the preferable smart home solution from the simple user interface.

2.8.2 Strengths

Xiaomi Home application is one of the mature IoT applications that support its own IoT products. There are a lot of features supported in the application such as device sharing, voice service, centralise control, automation, historical sensor data viewing and other awesome features. Xiaomi Home app is really powerful and user friendly for users to customise their own solution in the smart home domain. One of the strengths of the Xiaomi Home app is the quick and easy setup to add a gadget. For example, users can simply scan the QR code attached on the devices and finish the installation process in a few steps. Other than that, the automation feature of the application assists users in designing their smart home without heavy technical knowledge required.

2.8.3 Weaknesses

Xiaomi Home application faces a common weakness like some of the other IoT applications which is it only supports their own IoT products. Besides that, the Xiaomi Home application does not provide a guarantee of IoT sensor historical data to the users. For instance, if the users lost a sensor, all the related sensor data is not able to be retrieved by the users from the application. This may cause some severity issues for the users in the monitoring process.

2.9 Comparison between the existing application and proposed application

Table 1 shows the comparison between existing applications with proposed application..

Existing Products / **In App Features**	**Ezblock Studio**	**IoT in a box**	**SkyDrop Smart Sprinkler Controller**	**IoTool**	**Rasp Controller**	**Proposed Solution**
Quick and easy setup for non-tech-savvy users	No	Yes	No	No	No	Yes

Connect, monitor, and control multiple IoT devices with cloud based	No	Yes	No	Yes	No	Yes
Support multiple types of IoT sensors and actuators	Yes	Yes	No	Yes	Yes	Yes
Remote control the IoT sensors and actuators	Yes	No	Yes	Yes	Yes	Yes
Automated identification of connected IoT devices' components	No	Yes	Yes	No	No	Yes
Wireless connection	Yes	Yes	Yes	Yes	Yes	Yes
Real-time monitoring sensor data	Yes	Yes	Yes	Yes	Yes	Yes
Advanced data reporting (Eg, data history)	No	Yes	Yes	Yes	No	Yes
Automation rules setup from simple interface	No	Yes	Yes	Yes	No	Yes
Data guarantee (Replacement of old sensor data with new sensor)	No	No	No	No	No	Yes

Table 2.1 Comparison between the existing applications and proposed application (1)

Existing Products / In App Features	Lego Mindstorm Ev3 Programmer	SimpleLink SensorTag	Xiaomi Home	Proposed Solution
Quick and easy setup for non-tech-savvy users	Yes	Yes	Yes	Yes
Connect, monitor, and control multiple IoT devices with cloud based	No	No	Yes	Yes
Support multiple types of IoT sensors and actuators	Yes	Yes	Yes	Yes
Remote control the IoT sensors and actuators	Yes	No	Yes	Yes
Automated identification of connected IoT devices' components	Yes	Yes	Yes	Yes
Wireless connection	Yes	Yes	Yes	Yes
Real-time monitoring sensor data	Yes	Yes	Yes	Yes
Advanced data reporting (Eg, data history)	No	No	Yes	Yes
Automation rules setup from simple interface	No	No	Yes	Yes
Data guarantee (Replace old sensor data with new sensor)	No	No	No	Yes

Table 2.2 Comparison between the existing applications and proposed application (2)

2.10 Recommendations

After reviewing existing products, each of them has some limitations in terms of customisation, user-friendliness, controlling and monitoring IoT solutions. For instance, some products such as IoTool, Ezblock Studio, SimpleLink SensorTag and RaspController that advanced in customisation of IoT solutions are not user-friendly enough for those non-tech-savvy users to customise their own solutions. Users are expected to have a certain level of hardware and software knowledge regarding IoT devices. For instance, users may still need to configure extra settings when the devices are connected to the application, do some coding in the devices, and other related technical skills. At the same time, products that are user-friendly for users such as Skydrop and IoT in a Box may have some issues in customising IoT solutions. As both of them only support their own hardware products, future development may become a gap in enhancing customisation features of their products. Last but not least, some of these products also lack of remote and centralized control, data guarantee, simple automation rules setup, and advanced monitoring features to manage users' solutions.

Hence, the proposed application will be as user-friendly as possible to allow non-tech-savvy users to build their own IoT solutions easily. In the proposed application, users do not need to have any programming language or networking knowledge in setting up the IoT devices. Besides, this application will also support multiple types of common sensors that can be found in the market to enhance the customisation features as well. The automated identification of connected IoT devices' components by the application is one of the main strengths of the proposed application as compared to other existing applications. By using a small amount of cost, this feature will further enhance users' experience in enjoying a real plug and play feature provided by the application interface. Plug and play feature will play a vital role in assisting users to learn and customise their IoT solutions.

Other than that, simple automation rules setup is another highlight of the project. From the application, users are allowed to set conditions and actions to their preferable solution without heavy technical knowledge required. Furthermore, the other basic features will be included as well such as remote controlling, real-time monitoring, and notification service. Besides those basic monitoring and controlling features, the proposed application will also provide advanced reporting for the users to derive insight from data collected. As data is one of the important treasures in the IoT ecosystem, the proposed application will also ensure the data continuity when a sensor is lost or damaged in the future.

CHAPTER 3: SYSTEM DESIGN

The overall system developed consists of three parts: hardware, software and cloud. The hardware used in this system will be raspberry pi that adopting USB-enable sensor and actuator to support customisable of the solution. The development of raspberry pi and part of the cloud development is handled by another team member. In this report, the detailed system design will only cover software developed and cloud technology used.

3.1 System Design Diagram

3.1.1 System Architecture Diagram

First, IoT Core will act as a secure connection point to connect IoT devices in order to send data between the cloud platform and IoT devices. IoT Core can handle multiple IoT devices at the same time and ingest data from them. In this project, multiple Raspberry Pi are registered under one registry in IoT Core. After the data collected from IoT devices to the cloud, the data will pass to Cloud Pub/Sub service. There are few topics that will be created to collect sensor data, devices' status and other related devices data including a topic to monitor the connection between the devices and the cloud platform. Then, the data will be sent to the topic subscribers.

In the architecture diagram, cloud functions are the subscribers to receive messages from Cloud Pub/Sub and trigger code written. The codes will perform some backend logic on the data such as where to send the data and how to process the data. In this project, there are few important functions will be deployed in Cloud Function in order to save the heavy processing task on user mobile including sending data from Pub/Sub topic to Firebase Realtime database, limiting the number of child node in Firebase Realtime Database, performing analytic task on data collected, triggering automation and send notification to users, and also sending command back to the Raspberry Pi from the mobile application.

Furthermore, Cloud scheduler is used to trigger cloud function based on the schedule in order to perform centralise monitoring on the trigger set by users. Google Cloud Compute engine is used as a server to receive live video streaming from the Raspberry Pi camera module and send the video to mobile application users. Other than live video streaming, the camera module of Raspberry pi will capture a photo every day and store the photo to Cloud Storage for future usage.

Last but not least, Firebase Realtime Database will act as a database system for this project. There are various types of information stored in the database such as users' information, devices' information, triggers information and telemetry data. One of the main reasons for choosing the Firebase Realtime Database is the real-time nature of the database. Some data updates such as telemetry data in the database will directly reflect in the mobile application by the implementation of active listeners on the database. The ease of implementation of Firebase Realtime Database in Android application also saves the development time as well.

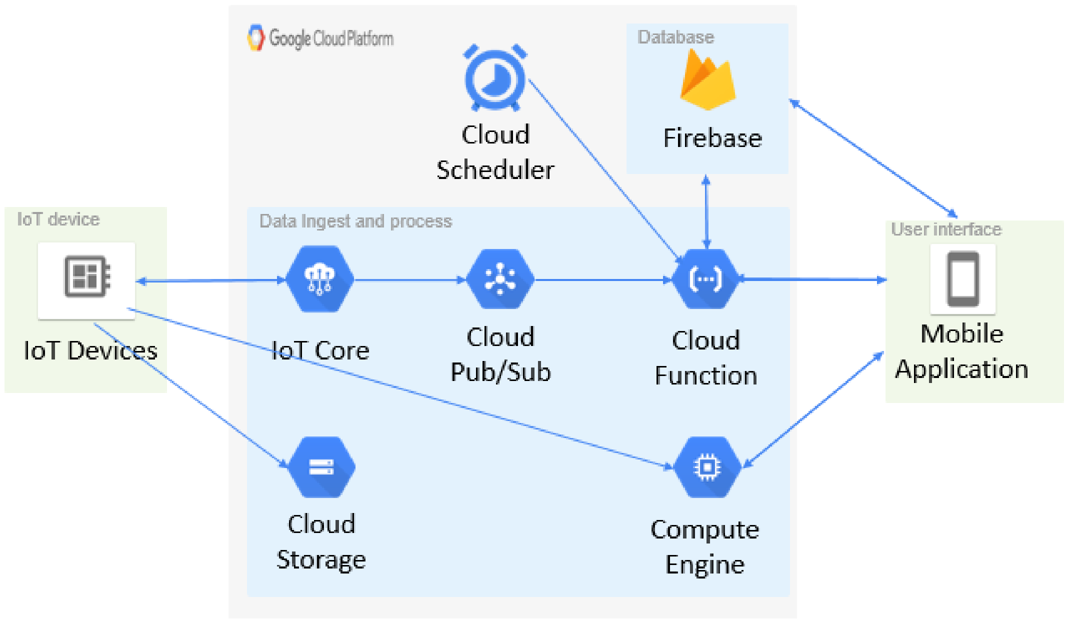


Figure 3.1 System Architecture Diagram

3.1.2 Use Case Diagram

Figure 3.2 shows a use case diagram for the overall system to demonstrate what users can do with the system. From the system, users are able to add and remove multiple devices. After adding or removing a device, the device dashboard will be updated accordingly. Furthermore, users can click into a particular device to view the device control panel to see which components are connected to the device. From the control panel, users can control the actuator connected, visualize real-time sensor data, view last week historical data, and view live video streaming. The control panel will be updated dynamically when the users remove or add a sensor/actuator from the devices. Moreover, users can set trigger rules from the application interface in order to automate their solution.

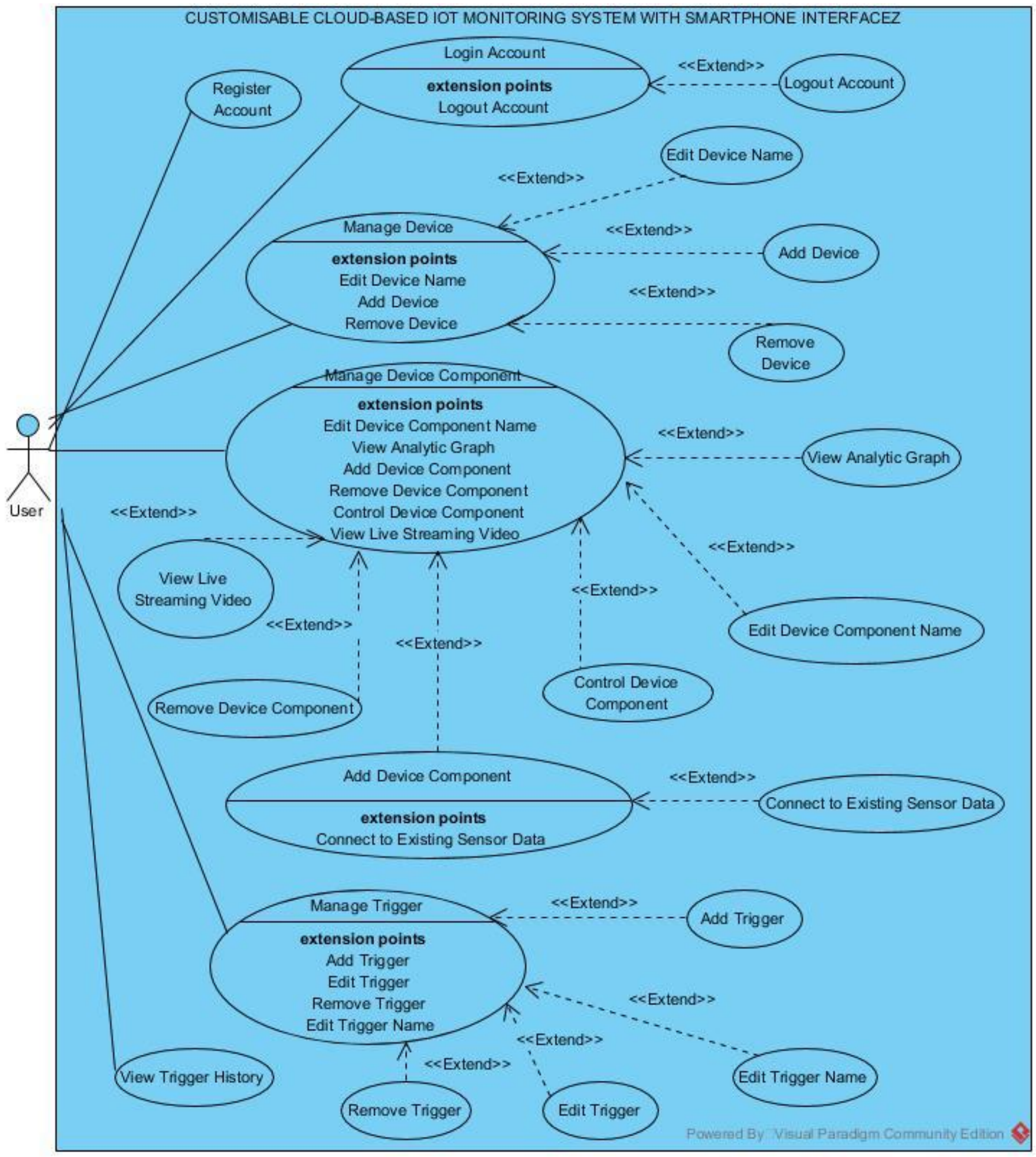


Figure 3.2 Use Case Diagram

3.1.3 Use Case Description

3.1.3.1 Use Case Description of Register Account

Use Case ID	UC001	**Version**	1.0
Use Case	Register Account		
Purpose	To register a user account.		

Actor	User	
Trigger	User launches the application.	
Precondition	User is not logged in.	
Scenario Name	**Step**	**Action**
Main Flow	1	User inputs valid Email and password.
	2	User presses Register button.
	3	System validates user email and password.
	4	System registers the user and redirects to Manage Device page. (MainHome activity).
Alternate Flow – Invalid Email or password	1.1.1	User inputs invalid Email or password.
	1.1.2	System validates user email and password.
	1.1.3	System displays error message.
Rules		-
Author		Ang Cheng Kee

Table 3.1 Use Case Description of Register Account

3.1.3.2 Use Case Description of Login Account

Use Case ID	UC002	**Version**	1.0
Use Case	Login Account		
Purpose	To login a user account.		
Actor	User		
Trigger	User launches the application.		
Precondition	User is not logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User inputs valid Email and password.	
	2	User presses Login button.	
	3	System validates user email and password.	
	4	System login the user and redirects to Manage Device Page. (MainHome activity).	
Alternate Flow – Invalid Email or Password	1.1.1	User inputs invalid Email or password.	
	1.1.2	System validates user email and password.	
	1.2.3	System displays error message.	

Rules	-
Author	Ang Cheng Kee

Table 3.2 Use Case Description of Login Account

3.1.3.3 Use Case Description of Logout Account

Use Case ID	UC003	**Version**	1.0
Use Case	Logout Account		
Purpose	To logout a user account from the application.		
Actor	User		
Trigger	User presses Logout button from drawer menu.		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User presses Logout button from drawer menu.	
	2	System logout the user and redirects to Login/Register page (MainActivity activity).	
Rules		-	
Author		Ang Cheng Kee	

Table 3.3 Use Case Description of Logout Account

3.1.3.4 Use Case Description of Manage Device

Use Case ID	UC004	**Version**	1.0
Use Case	Manage Device		
Purpose	To manage added IoT devices.		
Actor	User		
Trigger	User logins successfully into the application.		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	System display a list of added devices.	
	2	User view the list of added devices.	
Rules		-	
Author		Ang Cheng Kee	

Table 3.4 Use Case Description of Manage Device

3.1.3.5 Use Case Description of Add Device

Use Case ID	UC005	**Version**	1.0
Use Case	Add Device		
Purpose	To add an IoT device.		
Actor	User		
Trigger	User presses Add button from drawer menu, float button or bar menu**.**		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User presses Add button from drawer menu, float button or bar menu**.**	
	2	System requests for required phone services.	
	3	User allows the required services.	
	4	User presses Scan QR Code To Add Device button.	
	5	System displays camera interface.	
	6	User scans QR code attached on the device.	
	7	System retrieves information from the QR code.	
	8	System validates device connection via Bluetooth.	
	9	User inputs valid device password.	
	10	User presses Add button.	
	11	System validates device id and password.	
	12	System validates device Internet connectivity.	
	13	Users inputs valid WIFI name and WIFI password.	
	14	Users presses Add WIFI button.	
	15	System validates WIFI name and WIFI password.	
	16	System updates database and redirects to Manage Device page. (MainHome activity) .	
Alternate Flow – Services Denied	3.1	User denies the required services.	
	3.2	System redirects to Manage Device page. (MainHome activity)	
Alternate Flow – Scan Nearby Bluetooth Devices	4.1	Users presses Scan Nearby Bluetooth Devices button.	
	4.2	System displays nearby Bluetooth devices.	
	4.3	Users presses a displayed device from the Bluetooth device list.	
	4.4	Back to Main Flow Step 8.	

Alternate Flow – Connection Failure	8.1	System displays error message.
Alternate Flow - Invalid Device Password or Device is Added Already.	9.1	User inputs invalid device password.
	9.2	User presses Add button.
	9.3	System validates device id and password.
	9.4	System displays error message.
Alternate Flow – Have Internet Connection	13.1.1	System updates database and redirects to Manage Device page. (MainHome activity) .
Alternate Flow – No Internet Connection (Input with invalid WIFI name and WIFI password)	13.2.1	Users inputs invalid WIFI name and WIFI password.
	13.2.2	User presses Add WIFI Button
	13.2.3	System validates WIFI name and WIFI password.
	13.2.4	System displays error message.
Rules		-
Author		Ang Cheng Kee

Table 3.5 Use Case Description of Add Device

3.1.3.6 Use Case Description of Edit Device Name

Use Case ID	UC006	**Version**	1.0
Use Case	Edit Device Name		
Purpose	To edit added device’s name.		
Actor	User		
Trigger	User long presses a device from Manage Device page. (MainHome activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User long presses a device from Manage Device page. (MainHome activity).	
	2	User selects Edit Name from context menu.	
	3	System displays a dialog to request for device’s name.	
	4	User inputs device’s name.	
	5	User presses Yes button from the dialog.	

	6	System updates database and updates Manage Device page (MainHome activity).
Rules		-
Author		Ang Cheng Kee

Table 3.6 Use Case Description of Edit Device Name

3.1.3.7 Use Case Description of Remove Device.

Use Case ID	UC007	**Version**	1.0
Use Case	Edit Device Name		
Purpose	To remove a device.		
Actor	User		
Trigger	User long presses a device from manage Manage Device page. (MainHome activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User long presses a device from Manage Device page. (MainHome activity).	
	2	User selects Delete from context menu.	
	3	System updates database and Manage Device page. (MainHome activity).	
Rules		-	
Author		Ang Cheng Kee	

Table 3.7 Use Case Description of Remove Device

3.1.3.8 Use Case Description of Manage Device Component

Use Case ID	UC008	**Version**	1.0
Use Case	Manage Device Component		
Purpose	To manage added device’s components.		
Actor	User		
Trigger	User presses a device from Manage Device page. (MainHome activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	

Main Flow	1	User presses a device from Manage Device page. (MainHome activity).
	2	System displays a list of added device's sensor and actuator.
	3	User views the list of added device's sensor and actuator.
Rules		-
Author		Ang Cheng Kee

Table 3.8 Use Case Description of Manage Device Component

3.1.3.9 Use Case Description of Add Device Component

Use Case ID	UC009	**Version**	1.0
Use Case	Add Device Component		
Purpose	To add a sensor or actuator.		
Actor	User		
Trigger	User plugs in a sensor/actuator to a connected Raspberry pi.		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User plugs in a new sensor to a connected Raspberry pi.	
	2	System updates database.	
	3	System validates sensor/actuator information.	
	4	System displays a dialog to request for adding new device.	
	5	User presses Yes from dialog.	
	6	System updates Manage Device Component page (Dashboard activity).	
Alternate Flow – Plug In New/Existing Actuator	1.1.1	User plugs in a new/existing actuator to a connected Raspberry pi.	
	1.1.2	System validates sensor/actuator information.	
	1.1.3	System updates Manage Device Component page (Dashboard activity).	
Alternate Flow – Plug In Existing Sensor.	1.2.1	User plugs in an existing sensor to a connected Raspberry pi.	
	1.2.2	System validates sensor/actuator information.	
	1.2.3	System updates Manage Device Component page (Dashboard activity).	
Alternate Flow – Connect to Previous	5.1.1	User presses Connect to Previous Sensor Data from dialog.	
	5.1.2	System displays options of related sensor ID.	

Sensor Data (At least one related sensor ID)	5.1.3	User selects a sensor option and presses Done from dialog.
	5.1.4	System updates database and Manage Device Component page (Dashboard activity).
Alternate Flow – Connect to Previous Sensor Data (No related sensor ID)	5.2.1	User presses Connect to Previous Sensor Data from dialog.
	5.2.2	System displays error message.
	5.2.3	System updates Manage Device Component page (Dashboard activity).
Rules		-
Author		Ang Cheng Kee

Table 3.9 Use Case Description of Add Device Component

3.1.3.10 Use Case Description of Edit Device Component Name

Use Case ID	UC010	**Version**	1.0
Use Case	Edit Device Component Name		
Purpose	To edit sensor/actuator name.		
Actor	User		
Trigger	User long presses a device component from Manage Device Component page. (Dashboard activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User long presses a device component from Manage Device Component page. (Dashboard activity).	
	2	User selects Edit Name from context menu.	
	3	System displays a dialog to request for device's component name.	
	4	User inputs device's component name.	
	5	User presses Yes button from the dialog.	
	6	System updates database and Manage Device Component page (Dashboard activity).	
Rules		-	
Author		Ang Cheng Kee	

Table 3.10 Use Case Description of Edit Device Component Name

3.1.3.11 Use Case Description of Remove Device Component

Use Case ID	UC011	**Version**	1.0
Use Case	Remove Device Component		
Purpose	To remove a sensor/actuator.		
Actor	User		
Trigger	User removes a sensor/actuator from Raspberry pi.		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User removes a sensor/actuator from Raspberry pi.	
	2	System updates database and Manage Device Component page (Dashboard activity).	
Rules		-	
Author		Ang Cheng Kee	

Table 3.11 Use Case Description of Remove Device Component

3.1.3.12 Use Case Description of Control Device Component

Use Case ID	UC012	**Version**	1.0
Use Case	Control Device Component		
Purpose	To control a sensor/actuator.		
Actor	User		
Trigger	User presses toggle button of an actuator from Manage Device Component page.(Dashboard activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User presses toggle button of an actuator from Manage Device Component page.(Dashboard activity).	
	2	System updates database and Manage Device Component page (Dashboard activity).	
Rules		-	
Author		Ang Cheng Kee	

Table 3.12 Use Case Description of Control Device Component

3.1.3.13 Use Case Description of View Live Streaming Video

Use Case ID	UC013	**Version**	1.0
Use Case	View Live Streaming Video		
Purpose	To view live streaming video from connected camera.		
Actor	User		
Trigger	User presses any camera from Manage Device Component page. (Dashboard activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User presses any camera from Manage Device Component page. (Dashboard activity).	
	2	System plays video from cloud.	
Rules	-		
Author	Ang Cheng Kee		

Table 3.13 Use Case Description of View Live Streaming Video

3.1.3.14 Use Case Description of View Analytic Graph

Use Case ID	UC014	**Version**	1.0
Use Case	View Analytic Graph		
Purpose	To view analytic graph from connected sensor.		
Actor	User		
Trigger	User presses any sensor from Manage Device Component page. (Dashboard activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User presses any sensor from Manage Device Component page. (Dashboard activity).	
	2	System displays analytic graph.	
Rules	-		
Author	Ang Cheng Kee		

Table 3.14 Use Case Description of View Analytic Graph

3.1.3.15 Use Case Description of Manage Trigger

Use Case ID	UC015	**Version**	1.0
Use Case	Manage Trigger		
Purpose	To manage trigger.		
Actor	User		
Trigger	User presses Trigger button from drawer menu.		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User presses Trigger button from drawer menu .	
	2	System displays a list of added triggers.	
	3	User views the list of added triggers.	
Rules		-	
Author		Ang Cheng Kee	

Table 3.15 Use Case Description of Manage Trigger

3.1.3.16 Use Case Description of Add Trigger

Use Case ID	UC016	**Version**	1.0
Use Case	Add Trigger		
Purpose	To add new trigger.		
Actor	User		
Trigger	User presses Add button from Trigger page. (Trigger activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User presses Add button from Trigger page. (Trigger activity).	
	2	User inputs trigger information.	
	3	User presses Done button.	
	4	System updates database and Trigger page (Trigger activity).	
Rules		-	
Author		Ang Cheng Kee	

Table 3.16 Use Case Description of Add Trigger

3.1.3.17 Use Case Description of Edit Trigger

Use Case ID	UC017	**Version**	1.0
Use Case	Edit Trigger		
Purpose	To edit existing trigger		
Actor	User		
Trigger	User presses a trigger from Trigger page (Trigger activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User presses a trigger from Trigger page (Trigger activity).	
	2	System displays related trigger information.	
	3	User edits the information.	
	4	User presses Done button.	
	5	System updates database and Trigger page (Trigger activity) interface.	
Rules		-	
Author		Ang Cheng Kee	

Table 3.17 Use Case Description of Edit Trigger

3.1.3.18 Use Case Description of Remove Trigger

Use Case ID	UC018	**Version**	1.0
Use Case	Remove Trigger		
Purpose	To delete existing trigger		
Actor	User		
Trigger	User long presses a trigger from Trigger page (Trigger activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User long presses a trigger from Trigger page (Trigger activity).	
	2	User selects Delete from context menu	
	3	System updates database and Trigger page (Trigger activity).	
Rules		-	
Author		Ang Cheng Kee	

Table 3.18 Use Case Description of Remove Trigger

3.1.3.19 Use Case Description of Edit Trigger Name

Use Case ID	UC019	**Version**	1.0
Use Case	Edit Trigger Name		
Purpose	To edit trigger name.		
Actor	User		
Trigger	User long presses a trigger from Trigger page (Trigger activity).		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User long presses a trigger from Trigger page (Trigger activity).	
	2	User selects Edit Name from context menu.	
	3	System displays a dialog to request for trigger name.	
	4	User inputs trigger name.	
	5	User presses Yes button from the dialog.	
	6	System updates database and Trigger page (Trigger activity).	
Rules		-	
Author		Ang Cheng Kee	

Table 3.19 Use Case Description of Edit Trigger Name

3.1.3.20 Use Case Description of View Trigger History

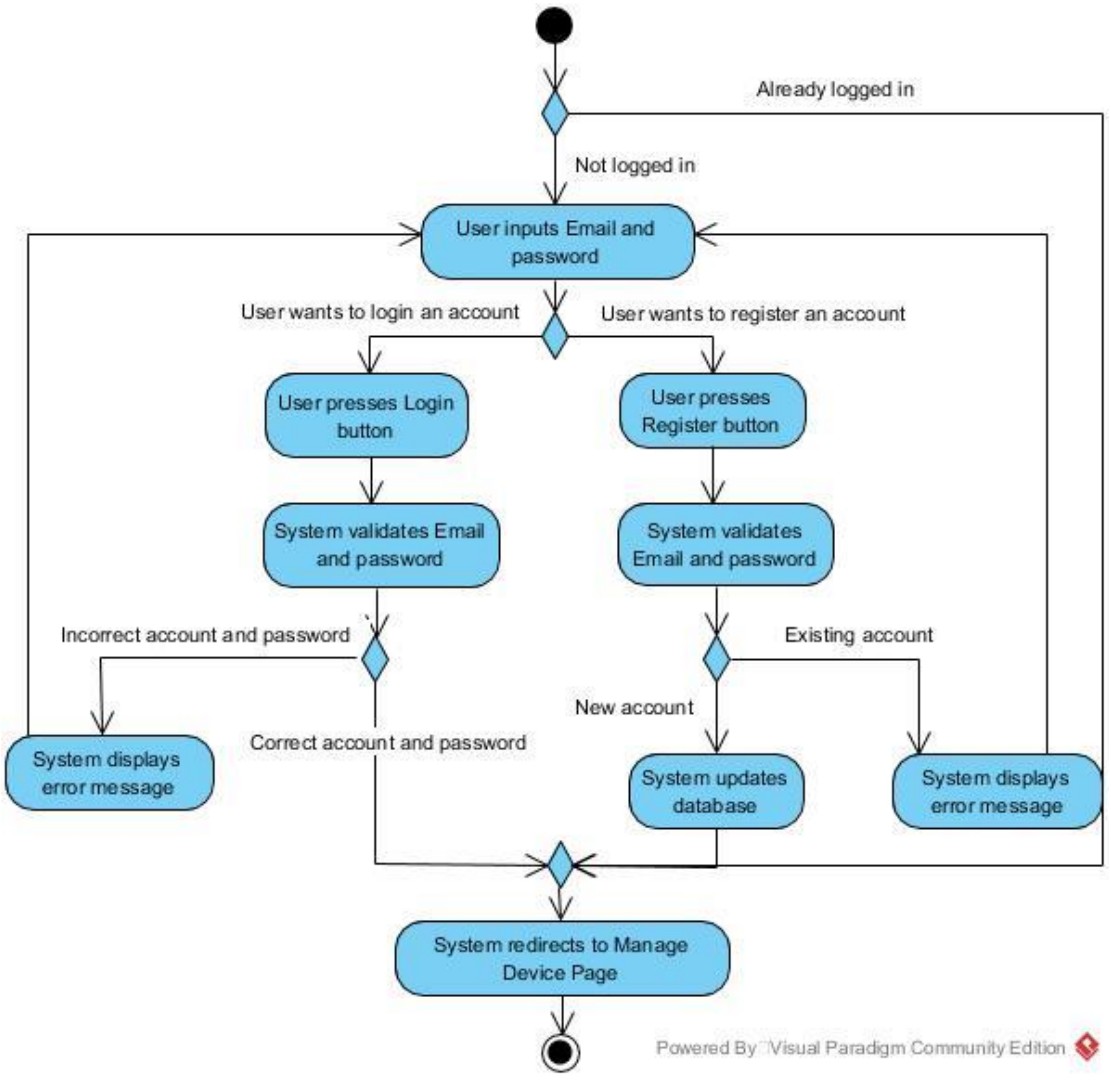
Use Case ID	UC0020	**Version**	1.0
Use Case	View Trigger History		
Purpose	To view trigger history.		
Actor	User		
Trigger	User presses Trigger Notification from drawer menu.		
Precondition	User is logged in.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User presses Trigger Notification from drawer menu.	
	2	System displays a list of trigger history.	
Rules		-	
Author		Ang Cheng Kee	

Table 3.20 Use Case Description of View Trigger History

3.1.4 Activity Diagram

3.1.4.1 Login/Register

This activity will be checked when users open the mobile application. If the user is already logged in, the Manage Device page will be displayed. Otherwise, the user needs to login his account or register a new account. First, the user is required to type in the email and password. If the user tends to login his account, click the Login button. If the user wants to register a new account, click the Register button. If the user clicks the Login button, the account and password will be validated. If the account and password are correct, the user will be navigated to the Manage Device page. Otherwise, an error message will be displayed. If the user clicks the Register button, the account will be validated. If it is a new account, the user will be navigated to the Manage Device page. Otherwise, an error message will be displayed.



Already logged in
Not logged in
User inputs Email and password
User wants to login an account
User wants to register an account
User presses Login button
User presses Register button
System validates Email and password
System validates Email and password
Incorrect account and password
Existing account
New account
Correct account and password
System displays error message
System updates database
System displays error message
System redirects to Manage Device Page
Powered By Visual Paradigm Community Edition

Figure 3.3 Activity Diagram of Login/Register

3.1.4.2 Logout Account

This activity will logout user account when the user presses the Logout button from drawer menu.

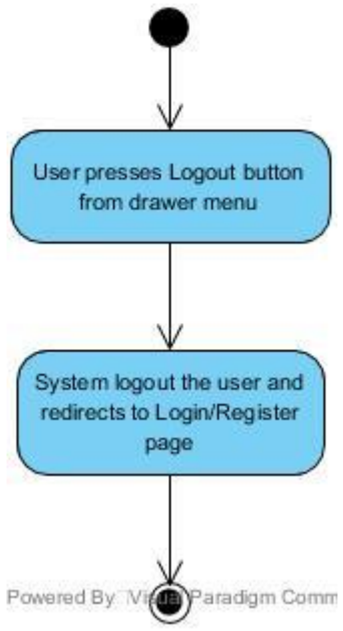

User presses Logout button from drawer menu
System logout the user and redirects to Login/Register page
Powered By Visual Paradigm Comm

Figure 3.4 Activity Diagram of Logout Account

3.1.4.3 Manage Device

After login the user account, user can manage all added devices via Manage Device page. The page will be the home screen of the application after user login. In this page, user can press a particular device in order to manage the device component. Besides, user can long press a device to edit or delete the device from the pop-out context menu. Moreover, user can add a device by pressing Add button from drawer menu, float button or bar menu**.**

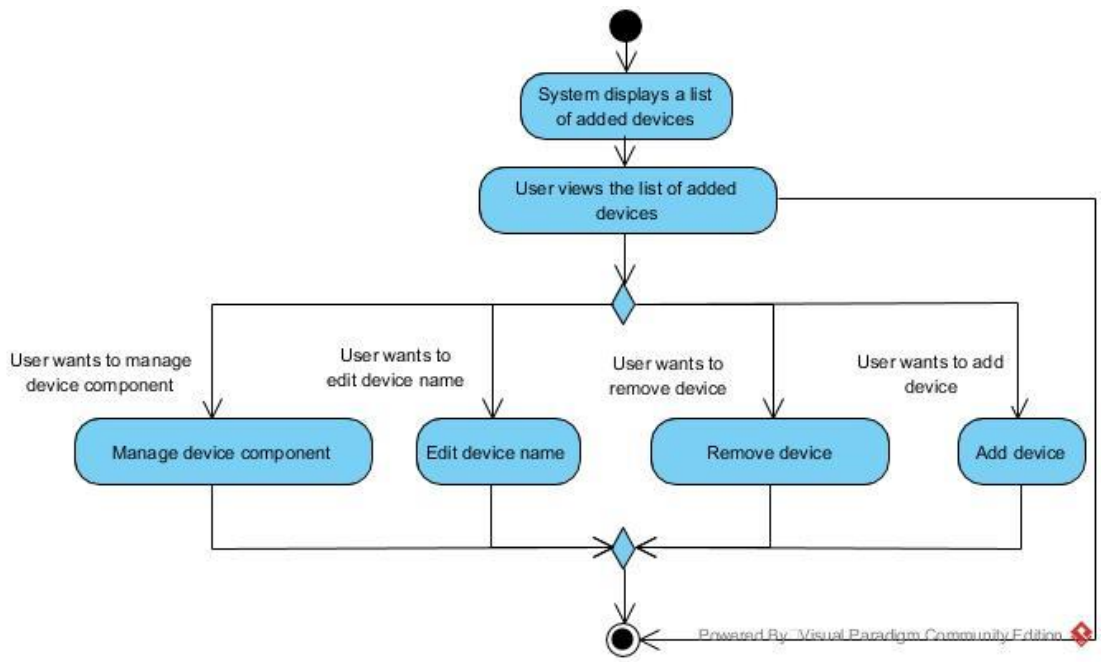

System displays a list of added devices
User views the list of added devices
User wants to manage device component
User wants to edit device name
User wants to remove device
User wants to add device
Manage device component
Edit device name
Remove device
Add device
Powered By: Visual Paradigm Community Edition

Figure 3.5 Activity Diagram of Manage Device

3.1.4.4 Add Device

When users want to connect to their IoT devices, they need to choose either to scan the QR code attached on the device or scan nearby devices via Bluetooth. Both methods require users to input the correct device password for security purpose. If the password is incorrect, an error message will show. Otherwise, the system will proceed to validate of device Internet connection. If the device does not have an Internet connection, the system will request for Wifi name and Wifi password in order to set an Internet connection for the devices. If the device has an Internet connection, the device will be added successfully without prompt for Wifi setting.

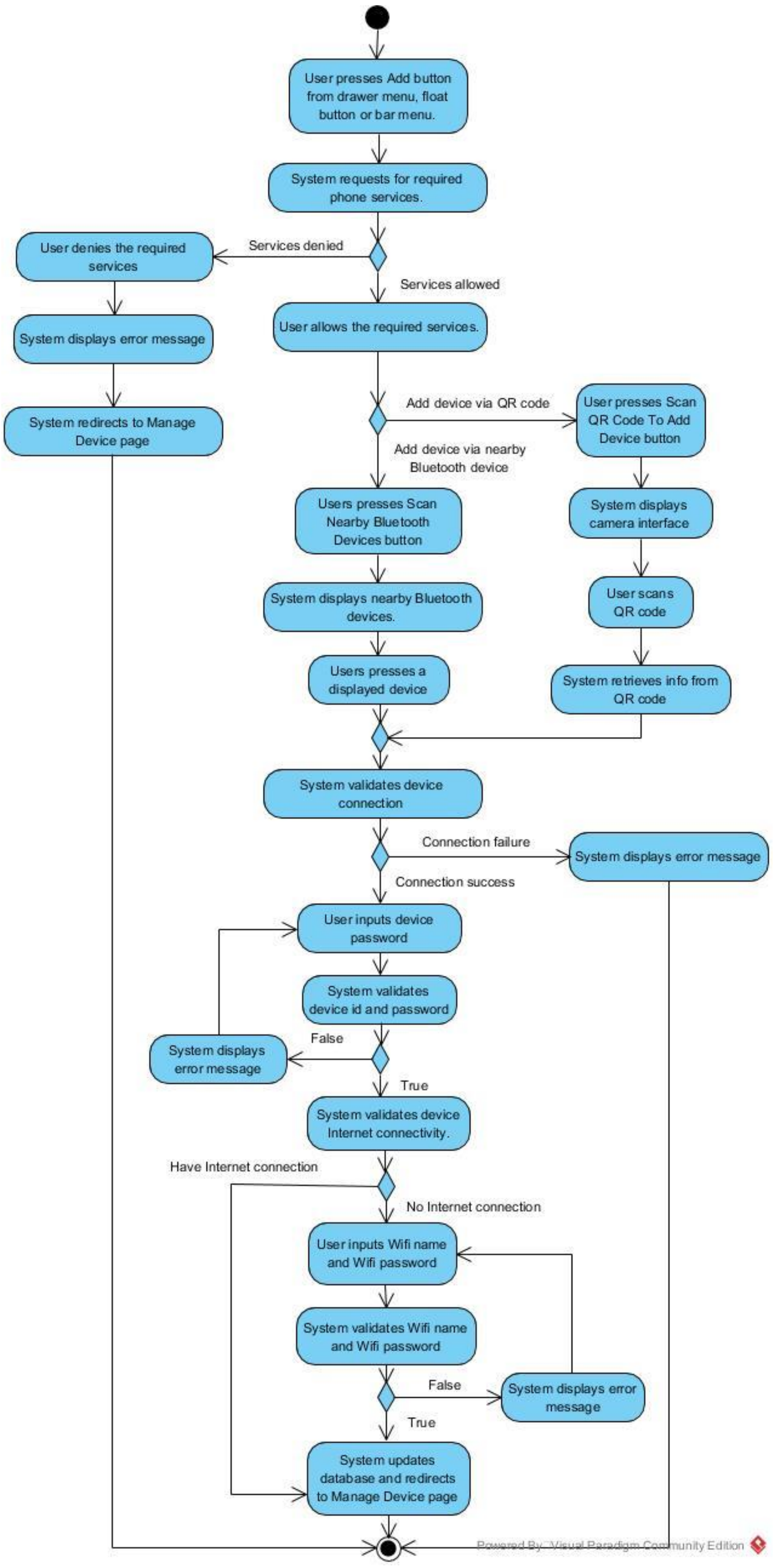

User presses Add button from drawer menu, float button or bar menu.
System requests for required phone services.
Services denied
User denies the required services
Services allowed
User allows the required services.
System displays error message
System redirects to Manage Device page
Add device via QR code
User presses Scan QR Code To Add Device button
Add device via nearby Bluetooth device
Users presses Scan Nearby Bluetooth Devices button
System displays camera interface
System displays nearby Bluetooth devices.
User scans QR code
Users presses a displayed device
System retrieves info from QR code
System validates device connection
Connection failure
System displays error message
Connection success
User inputs device password
System validates device id and password
False
System displays error message
True
System validates device Internet connectivity.
Have Internet connection
No Internet connection
User inputs Wifi name and Wifi password
System validates Wifi name and Wifi password
False
System displays error message
True
System updates database and redirects to Manage Device page
Powered By: Visual Paradigm Community Edition

Figure 3.6 Activity Diagram of Add Device

3.1.4.5 Edit Device Name

After the device added, user can edit the device name by selecting Edit Name from the context menu of the device.

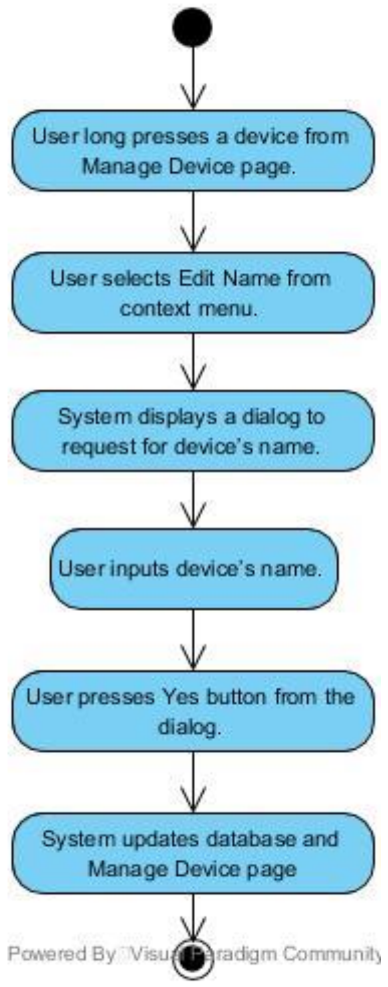

User long presses a device from Manage Device page.
User selects Edit Name from context menu.
System displays a dialog to request for device's name.
User inputs device's name.
User presses Yes button from the dialog.
System updates database and Manage Device page
Powered By Visual Paradigm Community

Figure 3.7 Activity Diagram of Edit Device Name

3.1.4.6 Remove Device

After the device added, user can remove a device by selecting Delete from the context menu of the device.

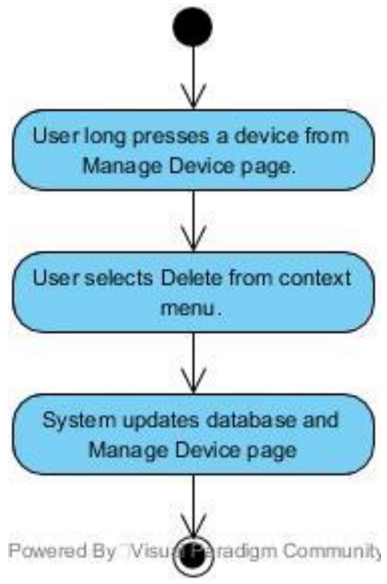

User long presses a device from Manage Device page.
User selects Delete from context menu.
System updates database and Manage Device page
Powered By Visual Paradigm Community

Figure 3.8 Activity Diagram of Remove Device

3.1.4.7 Manage Device Component

By pressing a device from the Manage Device page, users are allowed to manage the device's component via a dashboard that contains all the connected sensors or actuators in this device. In the dashboard, users can long press a sensor or an actuator panel to edit the component's name from a pop-out context menu. Besides, users can control the actuator by turning on/off the actuator from the dashboard. Moreover, users can also view live streaming video, add new sensor/actuator and remove existing sensor/actuator. Last but not least, users can press a sensor to view an analytic graph of the sensor data.

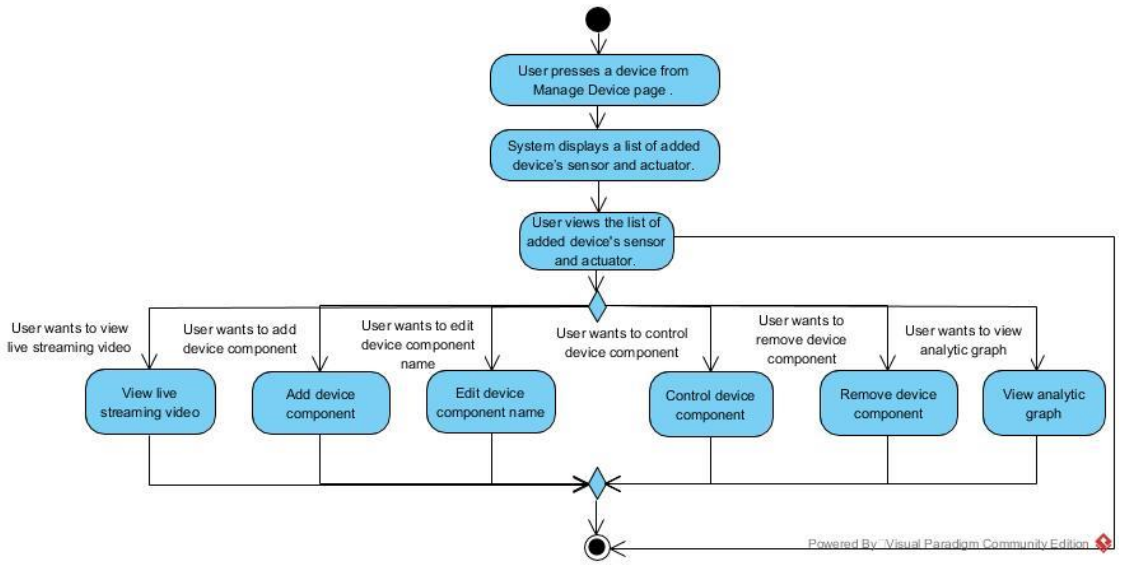

User presses a device from Manage Device page .
System displays a list of added device's sensor and actuator.
User views the list of added device's sensor and actuator.
User wants to view live streaming video
User wants to add device component
User wants to edit device component name
User wants to control device component
User wants to remove device component
User wants to view analytic graph
View live streaming video
Add device component
Edit device component name
Control device component
Remove device component
View analytic graph
Powered By Visual Paradigm Community Edition

Figure 3.9 Activity Diagram of Manage Device Component

3.1.4.8 Add Device Component

When users add a sensor or actuator to the device, the system will validate the plugged in component information. If users plug in an actuator or existing sensor, the Manage Device Component page will be updated. Otherwise (new sensor), a dialog box will show to the users. From the dialog box, users can choose to add this sensor as a new sensor or connect to previous sensor data. If users choose to add this sensor as a new sensor, the Manage Device Component page will be updated immediately. If users choose to connect to previous sensor data, the system will retrieve the available sensor data from the database. If there is at least 1 related data, the database and the Manage Device Component page will be updated. Otherwise, an error message will show to the users and the Manage Device Component page will be updated immediately as the sensor is treated as new sensor.

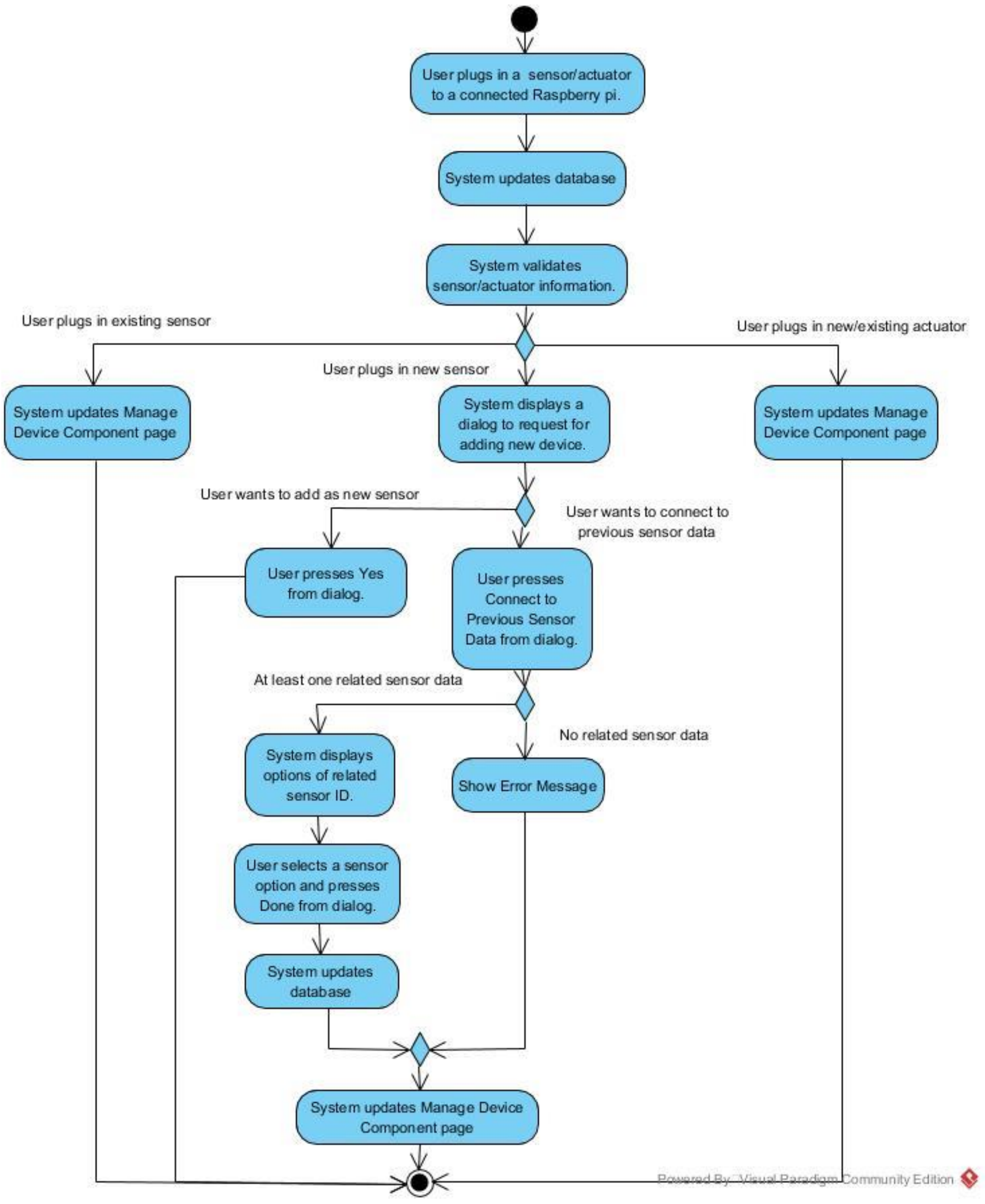

User plugs in a sensor/actuator to a connected Raspberry pi.
System updates database
System validates sensor/actuator information.
User plugs in existing sensor
User plugs in new sensor
User plugs in new/existing actuator
System updates Manage Device Component page
System displays a dialog to request for adding new device.
System updates Manage Device Component page
User wants to add as new sensor
User wants to connect to previous sensor data
User presses Yes from dialog.
User presses Connect to Previous Sensor Data from dialog.
At least one related sensor data
No related sensor data
System displays options of related sensor ID.
Show Error Message
User selects a sensor option and presses Done from dialog.
System updates database
System updates Manage Device Component page
Powered By Visual Paradigm Community Edition

Figure 3.10 Activity Diagram of Add Device Component

3.1.4.9 Edit Device Component Name

User can edit the device’s component name by selecting Edit Name from the context menu of the device component.

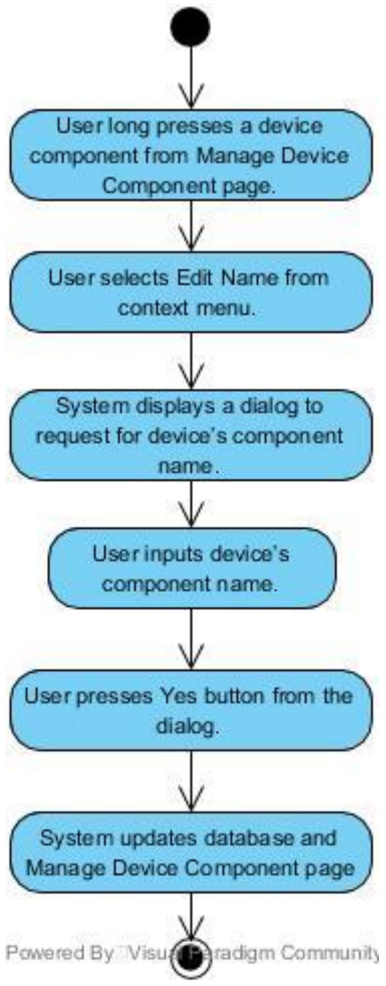

User long presses a device component from Manage Device Component page.
User selects Edit Name from context menu.
System displays a dialog to request for device's component name.
User inputs device's component name.
User presses Yes button from the dialog.
System updates database and Manage Device Component page
Powered By Visual Paradigm Community

Figure 3.11 Activity Diagram of Edit Device Component Name

3.1.4.10 Remove Device Component

When users remove a sensor or actuator from the device, database and Manage Device Component page will be updated.

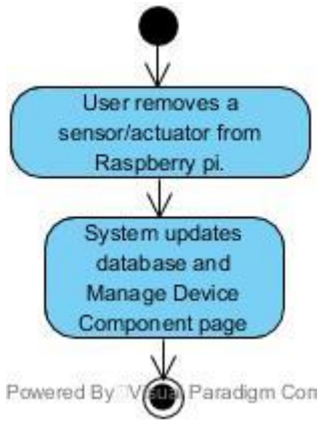

User removes a sensor/actuator from Raspberry pi.
System updates database and Manage Device Component page
Powered By Visual Paradigm Con

Figure 3.12 Activity Diagram of Remove Device Component

3.1.4.11 Control Device Component

User can turn on/off an actuator by pressing toggle button of the actuator from Manage Device Component page.

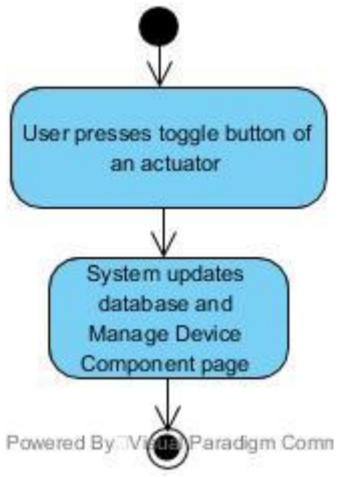

User presses toggle button of an actuator
System updates database and Manage Device Component page
Powered By Visual Paradigm Comn

Figure 3.13 Activity Diagram of Control Device Component

3.1.4.12 View Live Streaming Video

User can view live streaming video by pressing any camera from Manage Device Component page.

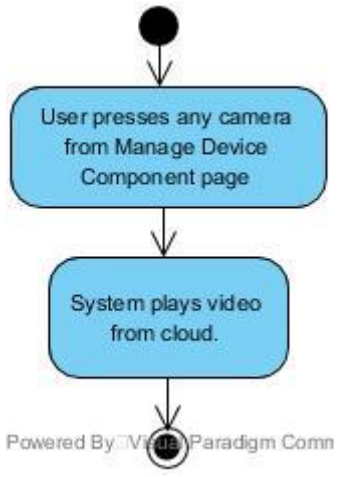

User presses any camera from Manage Device Component page
System plays video from cloud.
Powered By Visual Paradigm Comn

Figure 3.14 Activity Diagram of View Live Streaming Video

3.1.4.13 View Analytic Graph

User can view analytic graph by pressing any sensor from Manage Device Component page.

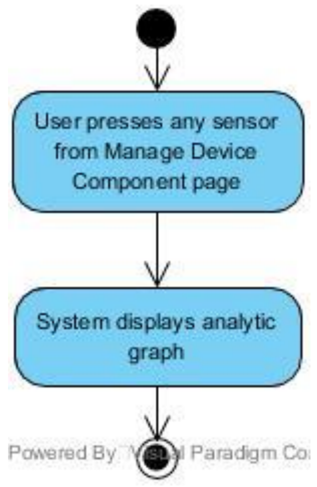

User presses any sensor from Manage Device Component page
System displays analytic graph
Powered By Visual Paradigm Co

Figure 3.15 Activity Diagram of View Analytic Graph

3.1.4.14 Manage Trigger

User can view a list of added triggers by pressing Trigger button from drawer menu. From Trigger page, user can also add, edit and remove trigger.

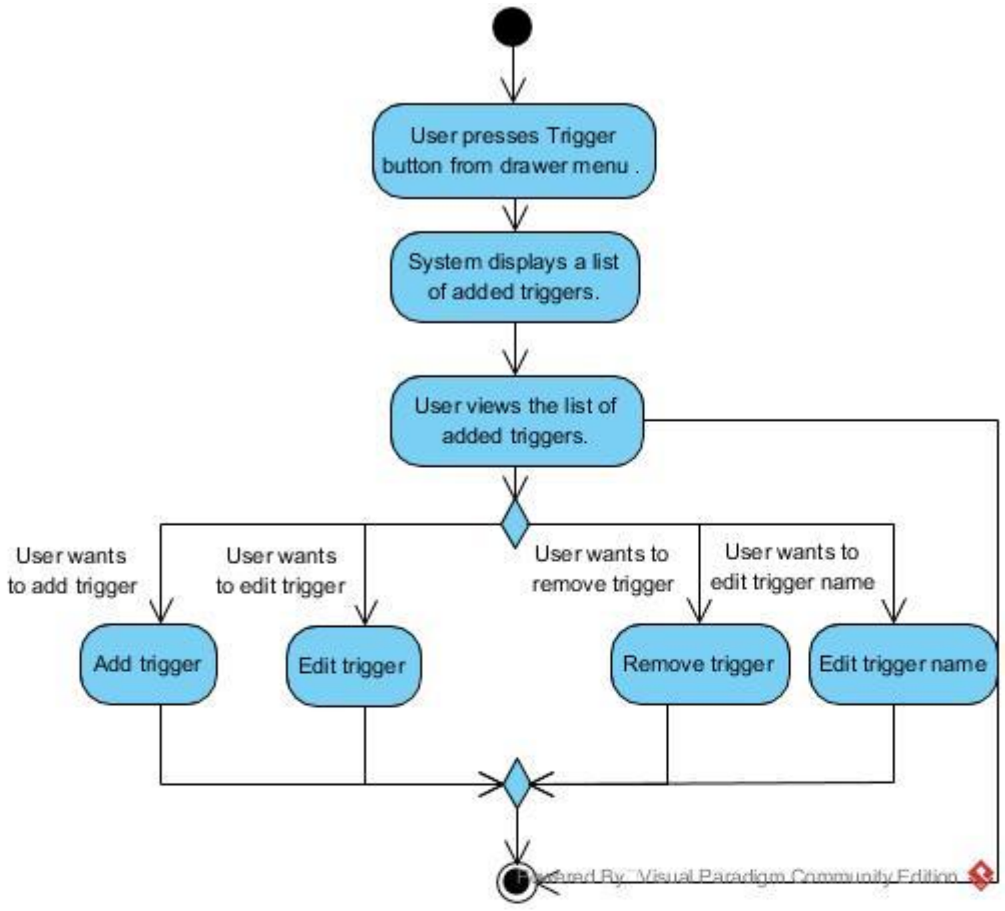

User presses Trigger button from drawer menu .
System displays a list of added triggers.
User views the list of added triggers.
User wants to add trigger
User wants to edit trigger
User wants to remove trigger
User wants to edit trigger name
Add trigger
Edit trigger
Remove trigger
Edit trigger name
Powered By Visual Paradigm Community Edition

Figure 3.16 Activity Diagram of Manage Trigger

3.1.4.15 Add Trigger

User can add a new trigger by pressing Add button from Trigger page.

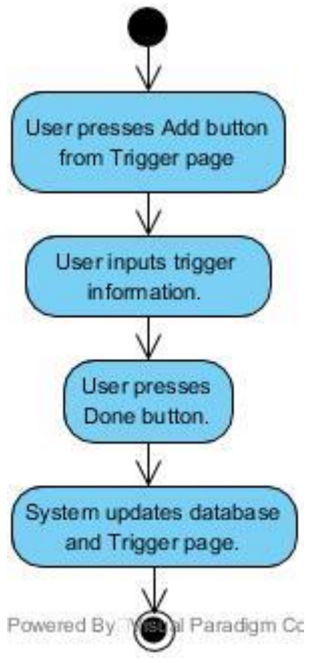

User presses Add button from Trigger page
User inputs trigger information.
User presses Done button.
System updates database and Trigger page.
Powered By Visual Paradigm Co

Figure 3.17 Activity Diagram of Add Trigger

3.1.4.16 Edit Trigger

User can edit added trigger information by pressing a trigger in Trigger page.

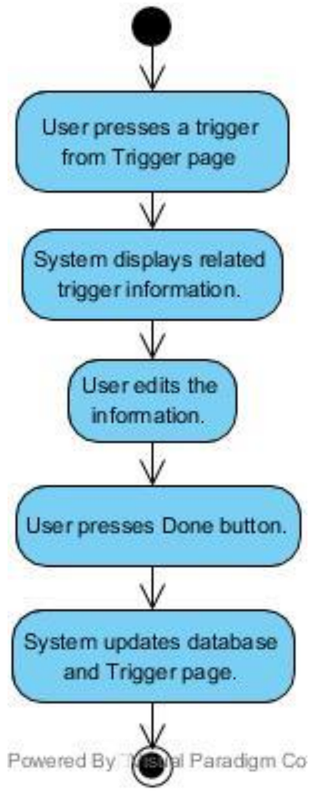

User presses a trigger from Trigger page
System displays related trigger information.
User edits the information.
User presses Done button.
System updates database and Trigger page.
Powered By Visual Paradigm Co

Figure 3.18 Activity Diagram of Edit Trigger

3.1.4.17 Remove Trigger

User can remove added trigger information by selecting Delete from context menu of a trigger.

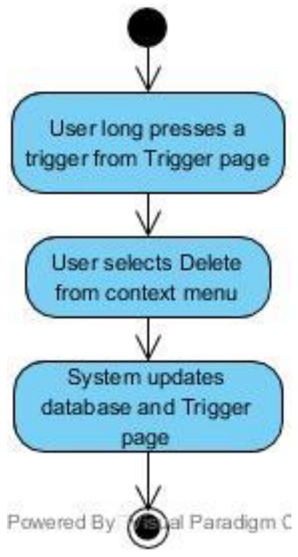

User long presses a trigger from Trigger page
User selects Delete from context menu
System updates database and Trigger page
Powered By Visual Paradigm C

Figure 3.19 Activity Diagram of Remove Trigger

3.1.4.18 Edit Trigger Name

User can edit added trigger name by selecting Edit Name from context menu of a trigger.

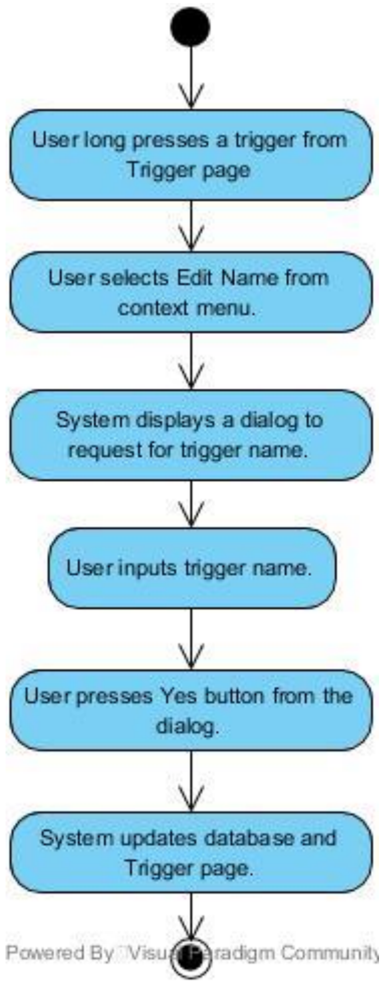

User long presses a trigger from Trigger page
User selects Edit Name from context menu.
System displays a dialog to request for trigger name.
User inputs trigger name.
User presses Yes button from the dialog.
System updates database and Trigger page.
Powered By Visual Paradigm Community

Figure 3.20 Activity Diagram of Edit Trigger Name

3.1.4.19 View Trigger History

User can view trigger history by pressing Trigger Notification from drawer menu.

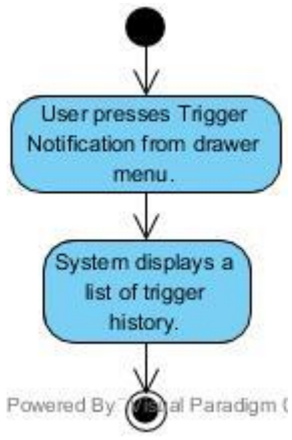

User presses Trigger Notification from drawer menu.
System displays a list of trigger history.
Powered By Visual Paradigm

Figure 3.21 Activity Diagram of View Trigger History

3.1.5 Entity Relationship Diagram

Figure 3.22 shows the entity relationship diagram of the system database. First, a device can have multiple things (sensor/actuator) connected whereas a thing can be shared among multiple devices. Besides, a device can be shared by many users and a user can have multiple devices. Moreover, a device may consist of multiple devices data collected by the things. A thing can collect more than one type of data. Other than that, a user can set multiple triggers that contain at least one condition and one action. Lastly, a user can view triggers' history of added triggers.

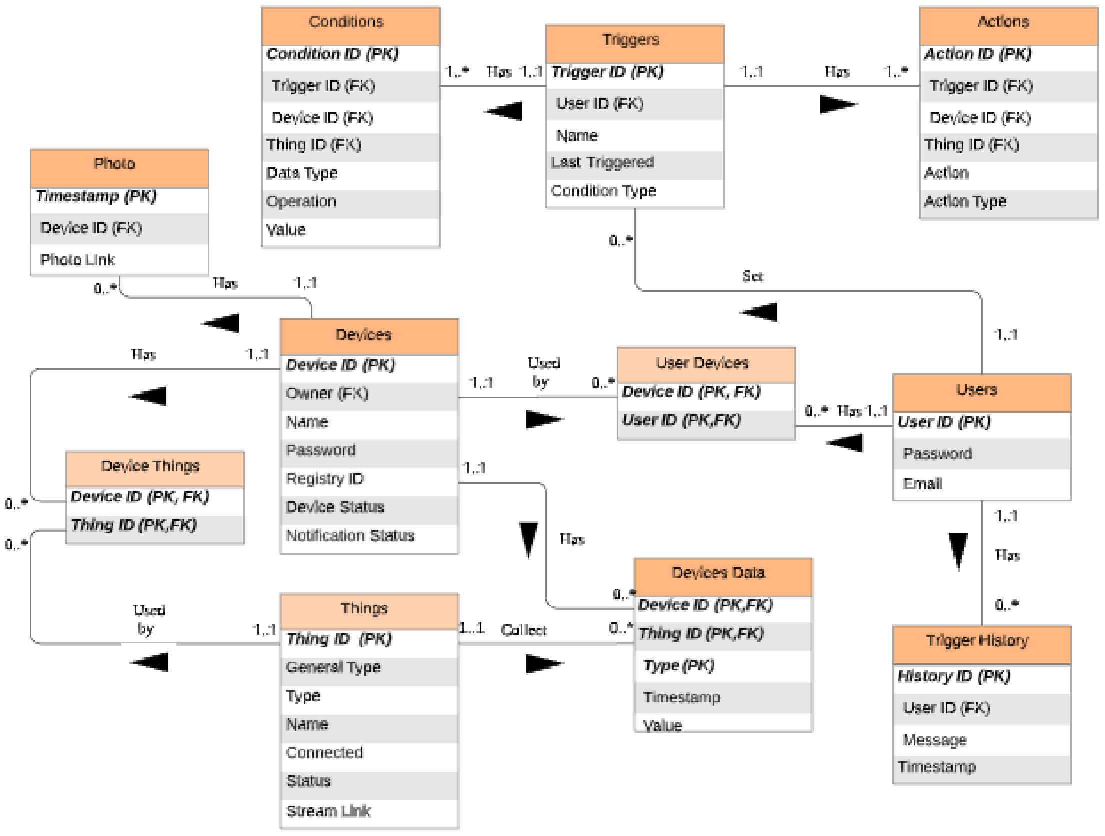
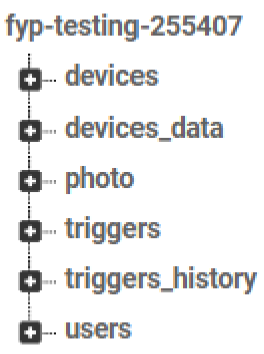


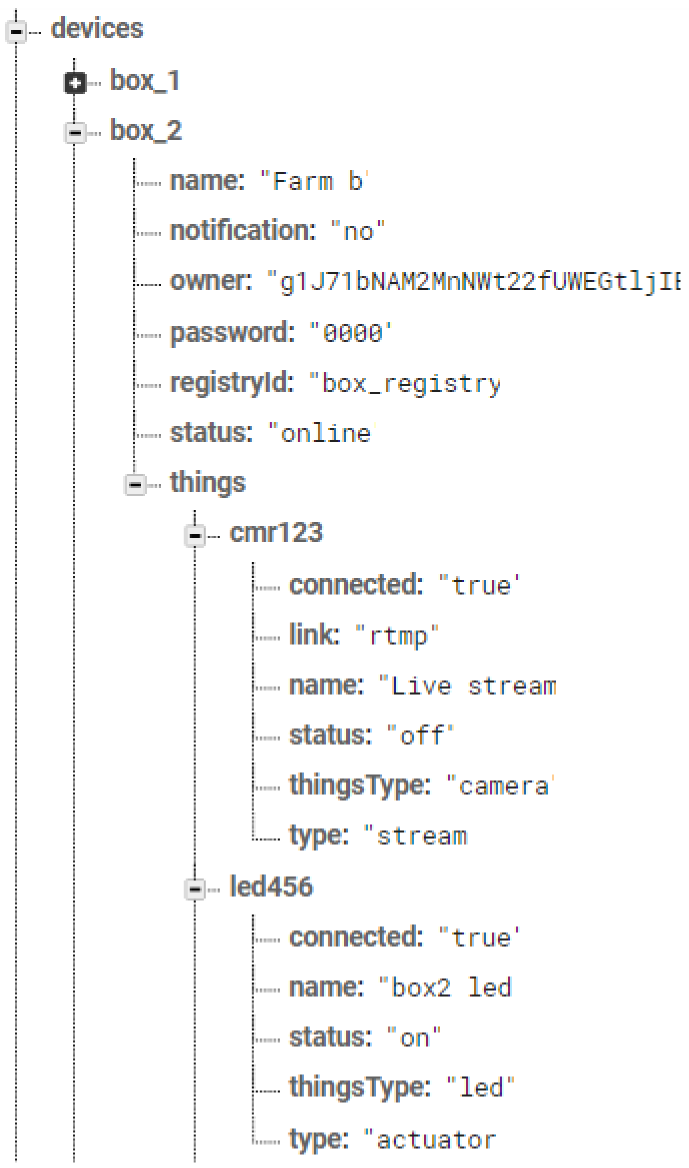
Figure 3.22 Entity Relationship Diagram

3.1.6 Data Structure of Firebase Realtime Database



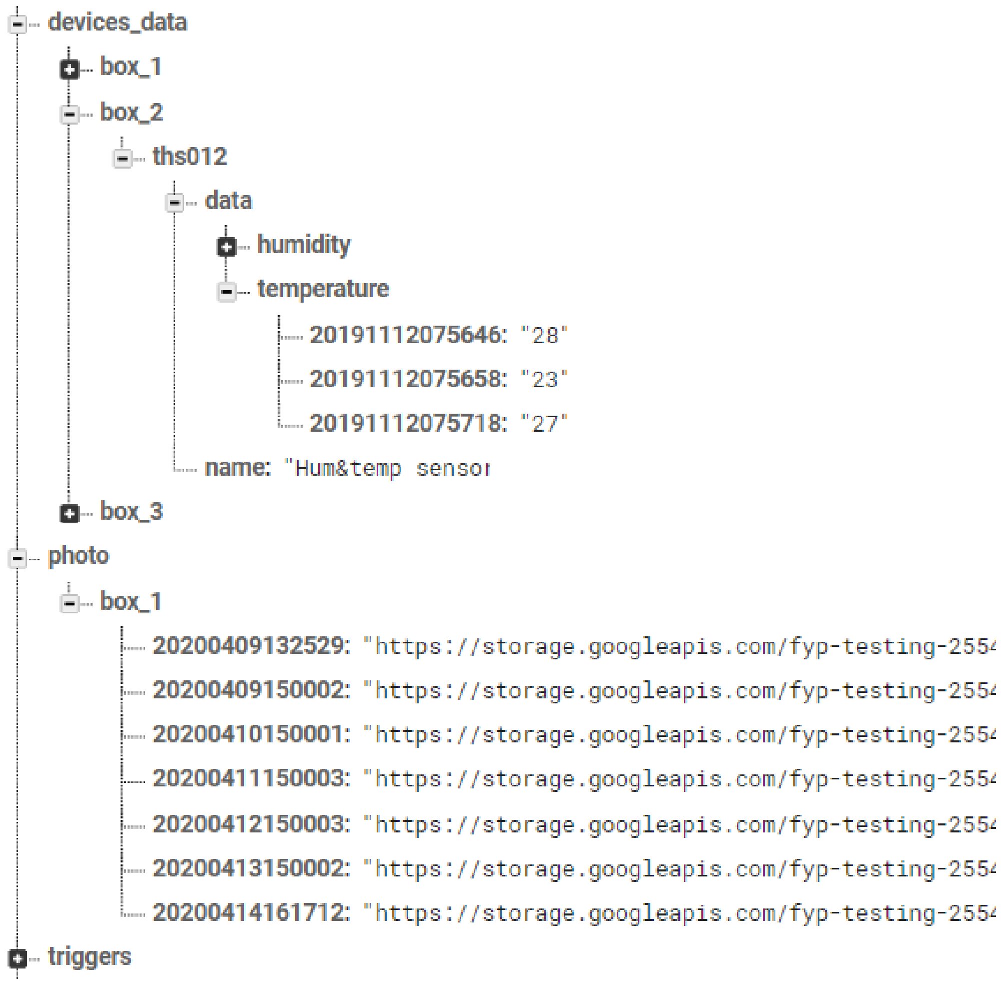
```
fyp-testing-255407
  devices
  devices_data
  photo
  triggers
  triggers_history
  users
```

Figure 3.23 Top Layer of Database Structure



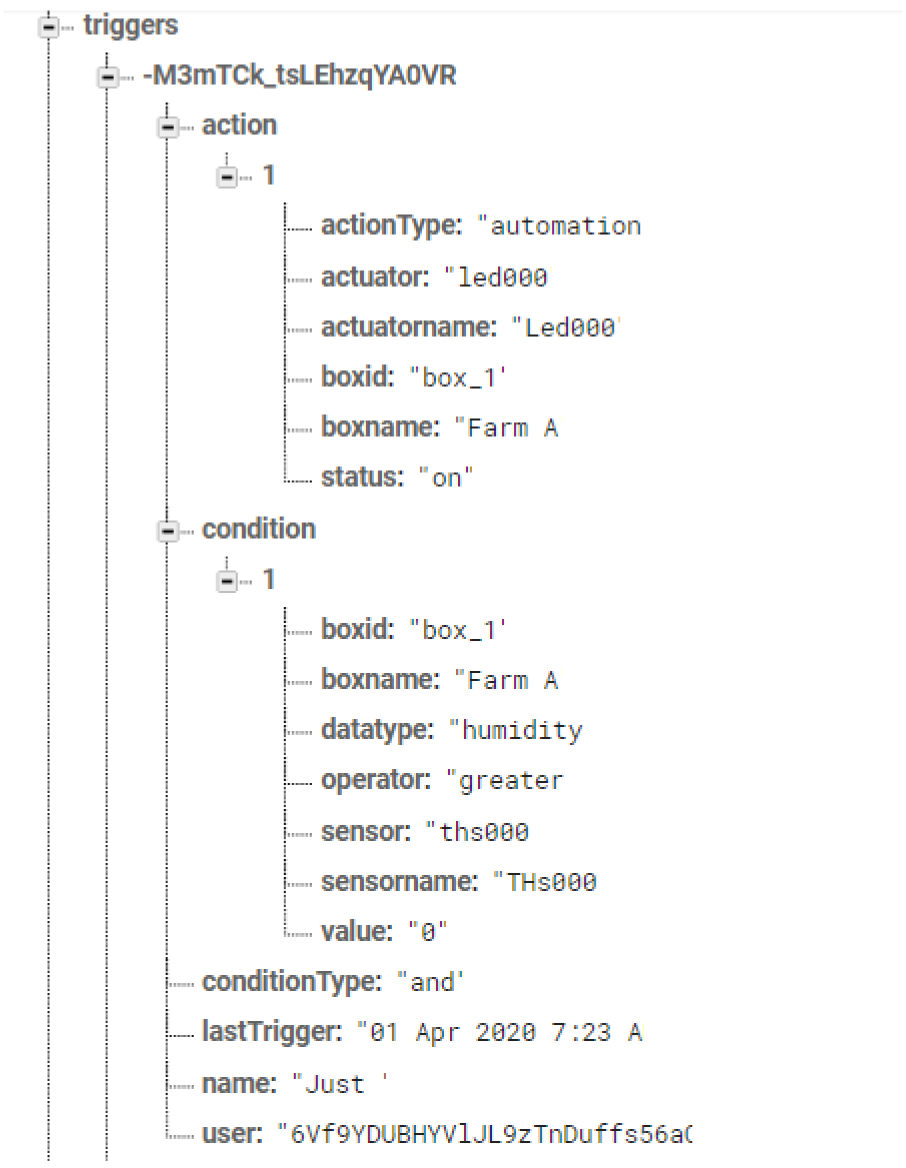
```
devices
  box_1
  box_2
    name: "Farm b'
    notification: "no"
    owner: "g1J71bNAM2MnNWt22fUWEGtljIE
    password: "0000'
    registryId: "box_registry
    status: "online
    things
      cmr123
        connected: "true'
        link: "rtmp"
        name: "Live stream
        status: "off'
        thingsType: "camera'
        type: "stream
      led456
        connected: "true'
        name: "box2 led
        status: "on"
        thingsType: "led"
        type: "actuator
```

Figure 3.24 Child Layer of Database Structure (1)



```
devices_data
    box_1
    box_2
        ths012
            data
                humidity
                temperature
                    20191112075646: "28"
                    20191112075658: "23"
                    20191112075718: "27"
            name: "Hum&temp sensor
    box_3
photo
    box_1
        20200409132529: "https://storage.googleapis.com/fyp-testing-2554
        20200409150002: "https://storage.googleapis.com/fyp-testing-2554
        20200410150001: "https://storage.googleapis.com/fyp-testing-2554
        20200411150003: "https://storage.googleapis.com/fyp-testing-2554
        20200412150003: "https://storage.googleapis.com/fyp-testing-2554
        20200413150002: "https://storage.googleapis.com/fyp-testing-2554
        20200414161712: "https://storage.googleapis.com/fyp-testing-2554
triggers
```

Figure 3.25 Child Layer of Database Structure (2)



```
triggers
  -M3mTCk_tsLEhzqYA0VR
    action
      1
        actionType: "automation
        actuator: "led000
        actuatorname: "Led000'
        boxid: "box_1'
        boxname: "Farm A
        status: "on"
    condition
      1
        boxid: "box_1'
        boxname: "Farm A
        datatype: "humidity
        operator: "greater
        sensor: "ths000
        sensorname: "THs000
        value: "0"
    conditionType: "and'
    lastTrigger: "01 Apr 2020 7:23 A
    name: "Just '
    user: "6Vf9YDUBHYVlJL9zTnDuffs56a(
```

Figure 3.26 Child Layer of Database Structure (3)

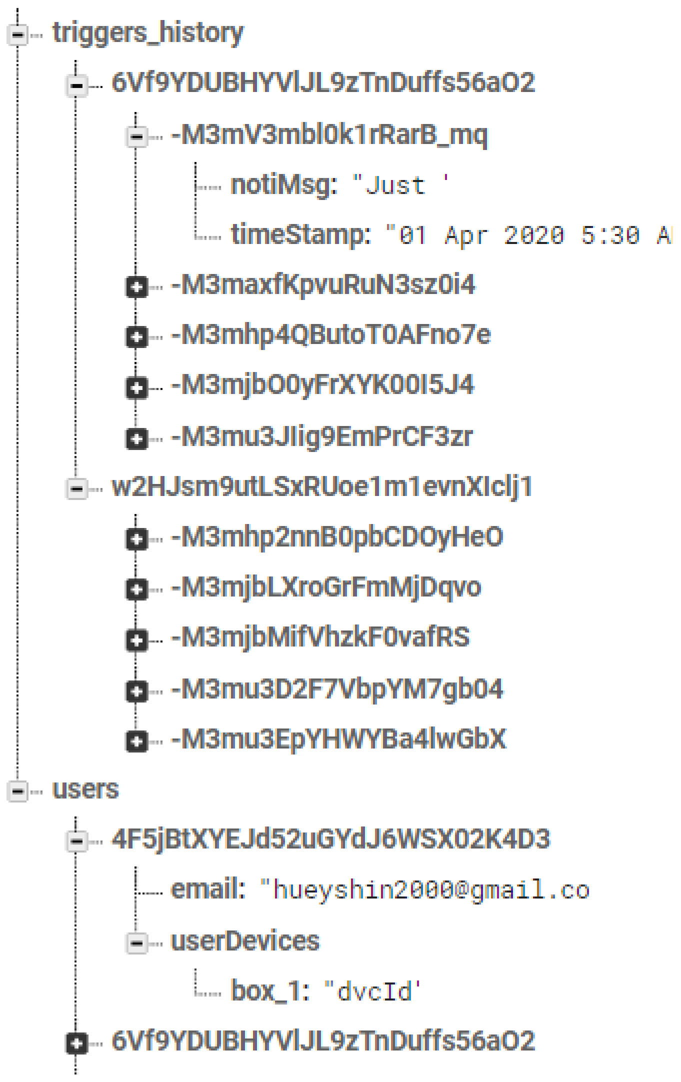

```
triggers_history
    6Vf9YDUBHYVlJL9zTnDuffs56aO2
        -M3mV3mbl0k1rRarB_mq
            notiMsg: "Just '
            timeStamp: "01 Apr 2020 5:30 A
        -M3maxfKpvuRuN3sz0i4
        -M3mhp4QButoT0AFno7e
        -M3mjbO0yFrXYK00I5J4
        -M3mu3JIig9EmPrCF3zr
    w2HJsm9utLSxRUoe1m1evnXIclj1
        -M3mhp2nnB0pbCDOyHeO
        -M3mjbLXroGrFmMjDqvo
        -M3mjbMifVhzkF0vafRS
        -M3mu3D2F7VbpYM7gb04
        -M3mu3EpYHWYBa4lwGbX
users
    4F5jBtXYEJd52uGYdJ6WSX02K4D3
        email: "hueyshin2000@gmail.co
        userDevices
            box_1: "dvcId'
    6Vf9YDUBHYVlJL9zTnDuffs56aO2
```

Figure 3.27 Child Layer of Database Structure (4)

CHAPTER 4: SYSTEM IMPLEMENTATION AND TESTING

4.1 System Methodology

The methodology for developing this project is Agile Development approach which consists of planning, analysis, design and implementation phases in an iterative cycle. The reason for choosing this methodology is due to its programming-centric model in order to shorten the software development time. Besides, the iterative model which develops the project incrementally able to enhance the flexibility of the project in order to adapt changes in the future. Other than that, the agile approach is also emphasising quick feedback from users by delivering software frequently to the customers. In this project, the software developed is improving incrementally as the supervisor's feedback will always aid in the software development process. Last but not least, agile development encourages extreme programming which requires at least two programmers perform pair programming. In this project, it consists of two members with the different project scope. One member handles hardware development whereas another member handles software development. During the development process, it requires both members involving in high frequency of discussion regarding development direction and technical issues.

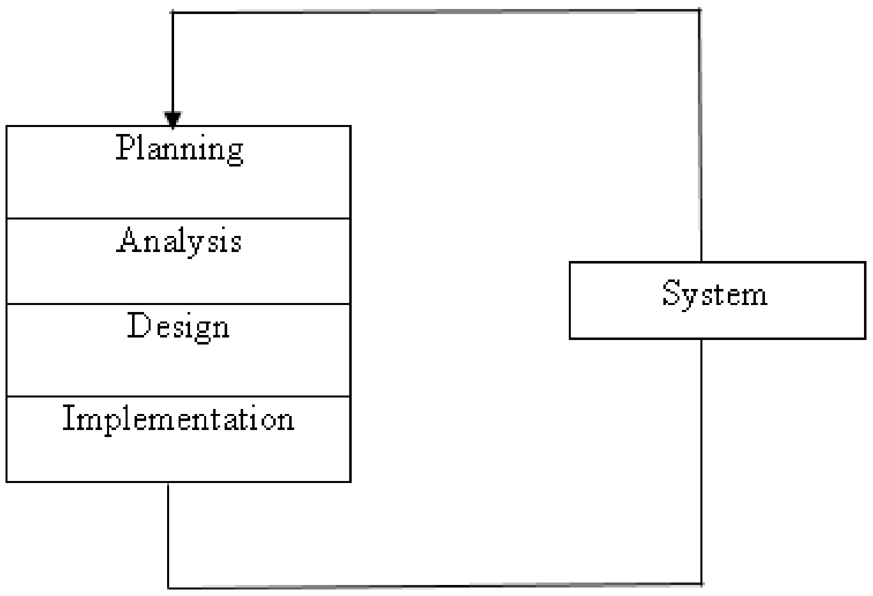


Figure 4.1 Agile Development Methodology

4.2 Project Workflow in Agile Development

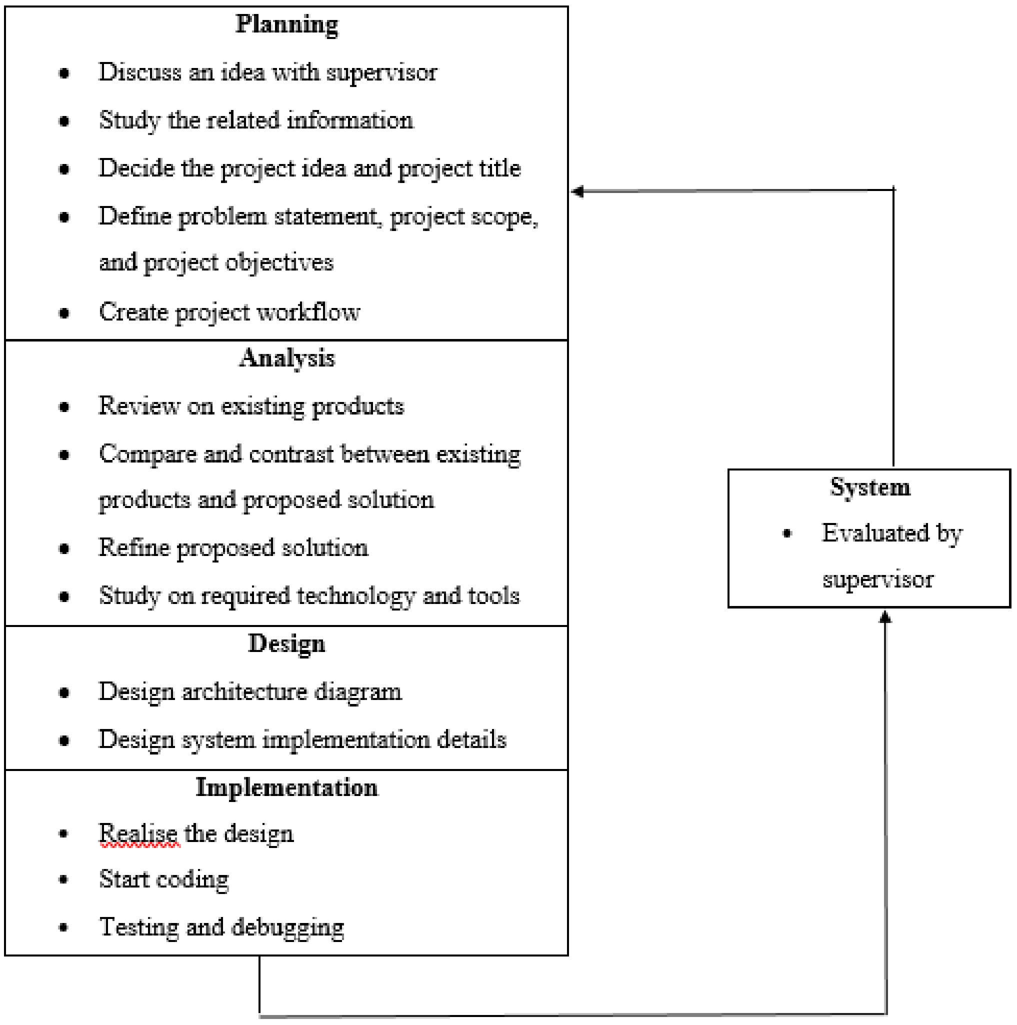

Planning
Discuss an idea with supervisor
Study the related information
Decide the project idea and project title
Define problem statement, project scope, and project objectives
Create project workflow
Analysis
Review on existing products
Compare and contrast between existing products and proposed solution
Refine proposed solution
Study on required technology and tools
Design
Design architecture diagram
Design system implementation details
Implementation
Realise the design
Start coding
Testing and debugging
System
Evaluated by supervisor

Figure 4.2 Project Workflow in Agile Development

4.2.1 Planning

First, an idea regarding IoT products was proposed and discussed with the supervisor. In the discussion, the problem domain and the project contribution were discussed to define the significant value of the project. After the discussion, some of the studies regarding the IoT domain were carried out to further analyse the requirement. After researching, the project idea was decided and a project title was created. Then, problem statement, project objectives and project scope were defined and a brief workflow of the project was designed and created as well.

4.2.2 Analysis

In this phase, a lot of research on similar products had been carried out in order to have a better understanding of how were existing solutions performed in solving IoT problems. In this project, these existing products were studied in terms of their features, operating processes, strengthens and weaknesses. After reviewing these products, a table of comparison between the products and the proposed solution was created. In this table, essential and advance IoT application features were compared in the aspect of user-friendliness, customisable, monitoring, and controlling IoT. After comparison, the proposed solution was refined a bit with some added-value features to be developed in the future. During the analysis phase, related tools and technologies were also part of the research in order to identify which tools and technology to be used in this project.

4.2.3 Design

In the design phase, an architecture diagram was designed to fulfil the project requirements. This architecture diagram shows the overall concept of the proposed system and the related cloud components to be used. There are 5 major components used for this system– IoT Core, Cloud Pub/Sub, Cloud Function, Cloud Scheduler, and Firebase. The detailed description was discussed in the system design section. Besides designing the architecture diagram, system implementation details were represented in this phase including use case diagram, use case description, activity diagram, entity relationship diagram and data structure of Firebase Realtime Database.

4.2.4 Implementation

In this phase, it is time to realise the designs mentioned in the design phase. First, start to configure the Google Cloud Platform as a customisable IoT platform to support IoT devices and the mobile application interface. Actual coding is also take place in this phase in order to come out with a real working product. During the coding period, testing and debugging process are carried out at the same time. At the end of this phase, a system is delivered to the supervisor to evaluate. If there is any error found or changes required in the system, the development cycle will repeat until reaching expectations.

4.3 Project Timeline

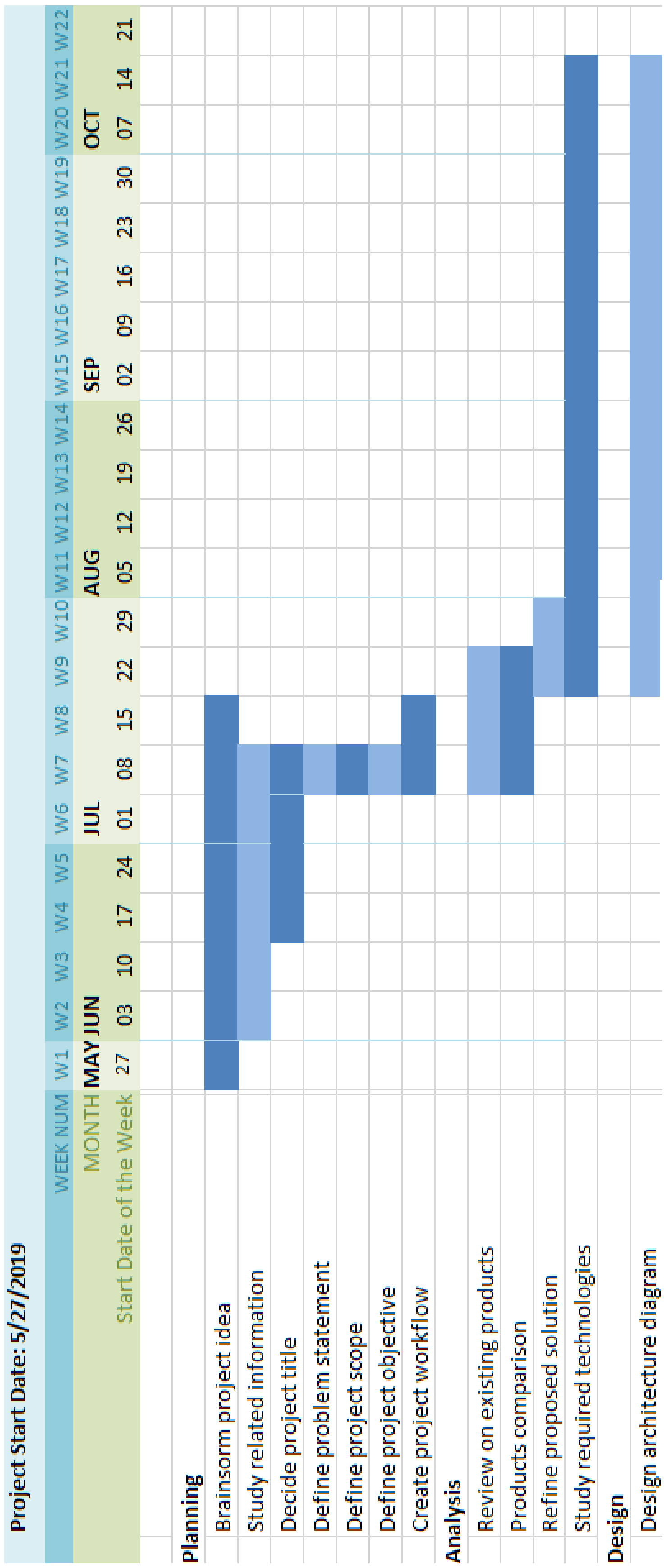


Figure 4.3 Gantt Chart (1)

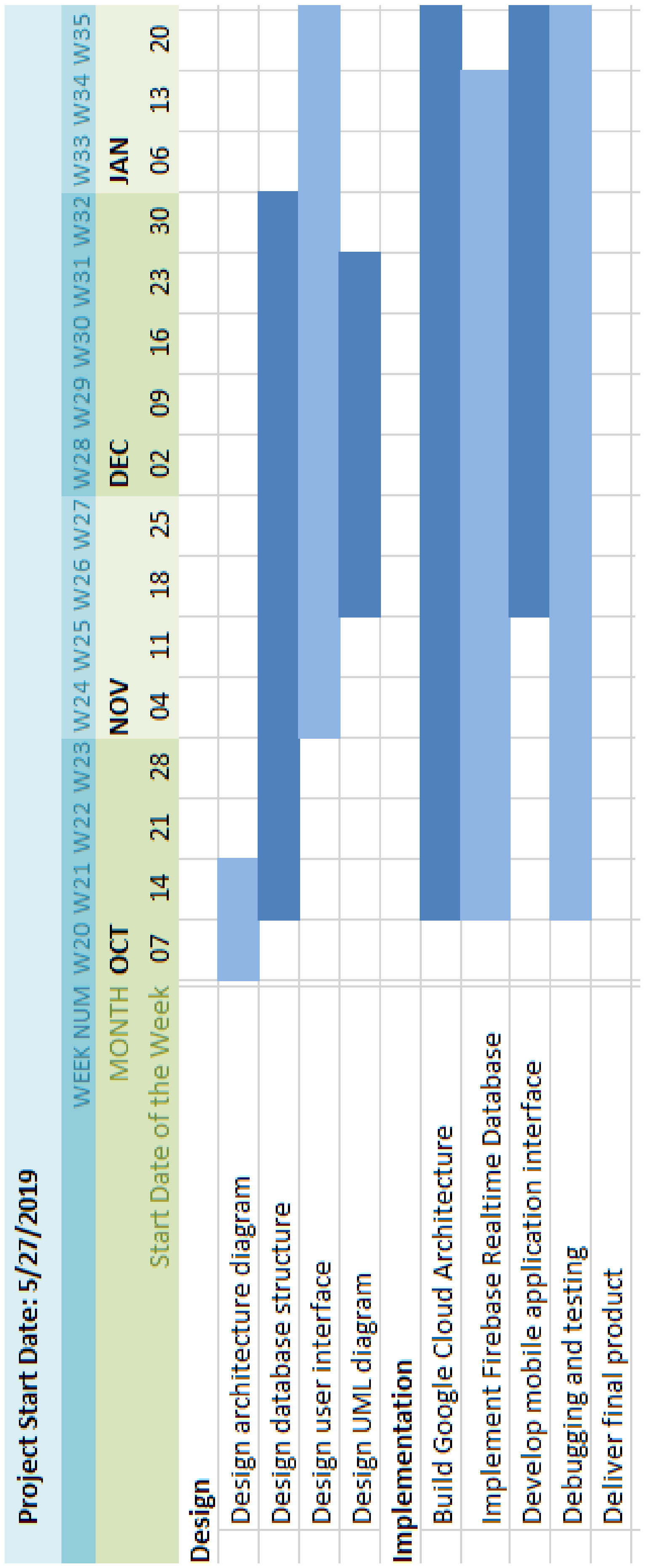


Figure 4.4 Gantt Chart (2)

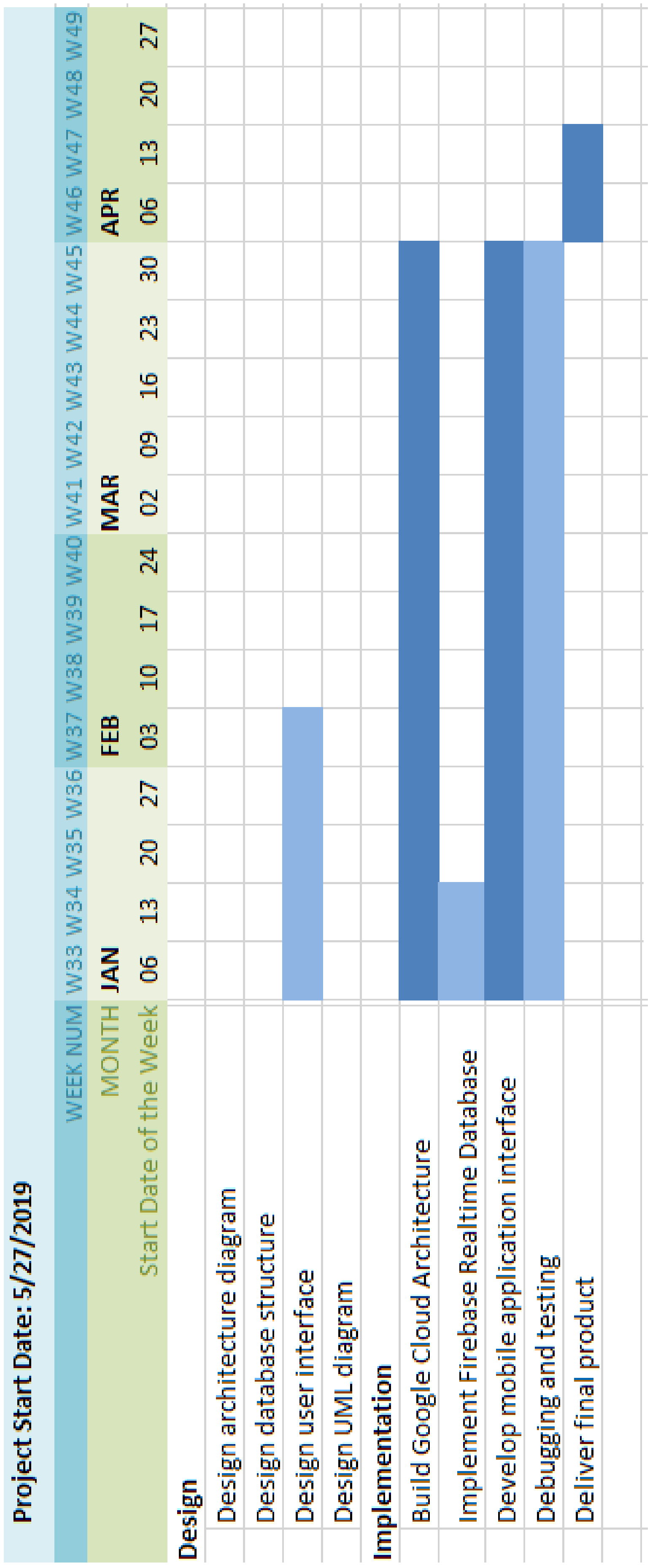


Figure 4.5 Gantt Chart (3)

4.4 Technologies and Tools Involved

There are a lot of technologies and tools involved in this projects including hardware, software, API, and Google Cloud platform. In this chapter, the details of hardware implementation will not be covered as it will be included in another team member’s report.

4.4.1 Raspberry Pi 3 Model B+

Raspberry Pi 3 Model B+ will be the main hardware device used in this project. It is a computer that supports customisation to develop own solution. The operating system used for the Raspberry Pi in this project is Raspbian Buster operating system. As the Raspberry Pi 3 Model B+ has a built-in wireless connection feature, it allows the devices to connect to nearby Wifi and the mobile application via Bluetooth protocol. Moreover, the Raspberry Pi in this project will act as a master device that controls sensors and actuators with the use of mini USB development board. Last but not least, Raspberry Pi will be the device that registered under Google IoT Core in Google Cloud Platform.

Figure 4.6 Raspberry Pi 3 Model B+

4.4.2 Digispark ATTINY85 Mini USB Development Board

The USB development board is a powerful programmable microcontroller which able to integrate with multiple types of sensors and actuators to form a USB-enabled sensor and actuator module. For instance, it can integrate with humidity and temperature sensor to achieve a plug and play feature in this project by using the USB port. The microcontroller can be considered as a middleman between the Raspberry Pi computer, the sensor and the actuator.

Figure 4.7 Digispark ATTINY85 Mini USB Development Board

4.4.3 DHT11 Temperature and Humidity Sensor

DHT11 temperature and humidity sensor is used to read the environment's humidity and temperature value.

Figure 4.8 DHT11 Temperature and Humidity Sensor

4.4.4 LDR Module

The role of LDR Module is a light intensity measurement sensor that provides reading based on the intensity of light.

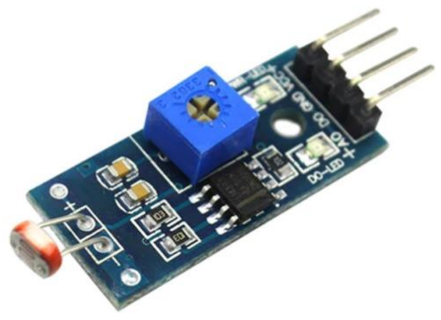


Figure 4.9 LDR Module

4.4.5 LED Board

The LED board used in this project is serving the purpose of providing light energy.

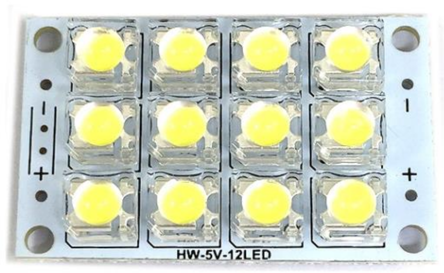


Figure 4.10 LED Board

4.4.6 Water sprinkler

A water pump that integrates with the mini USB development board as mentioned in the previous section will form a USB-enable actuator module as shown in figure 4.6. It is used to water the plant in the current project.

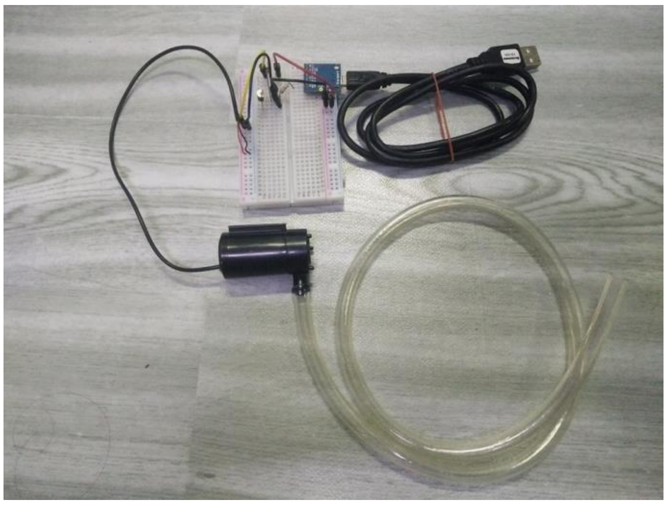


Figure 4.11 Integration of Water Sprinkler and Microcontroller

4.4.7 USB-Enabled Web Camera

A USB-enabled web camera is added to this project to allow users to view live streaming video from the camera. The camera will stream the video captured to a server hosted in Google Cloud Platform.

Figure 4.12 USB-enabled Web Camera

Figure 4.8 and figure 4.9 show the overall hardware implementations in this project.

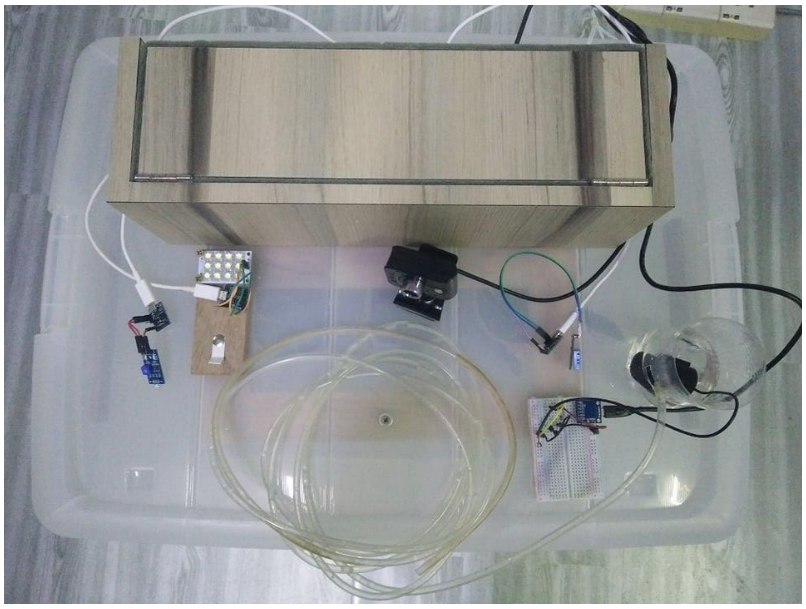


Figure 4.13 Overall Hardware Implementation (1)

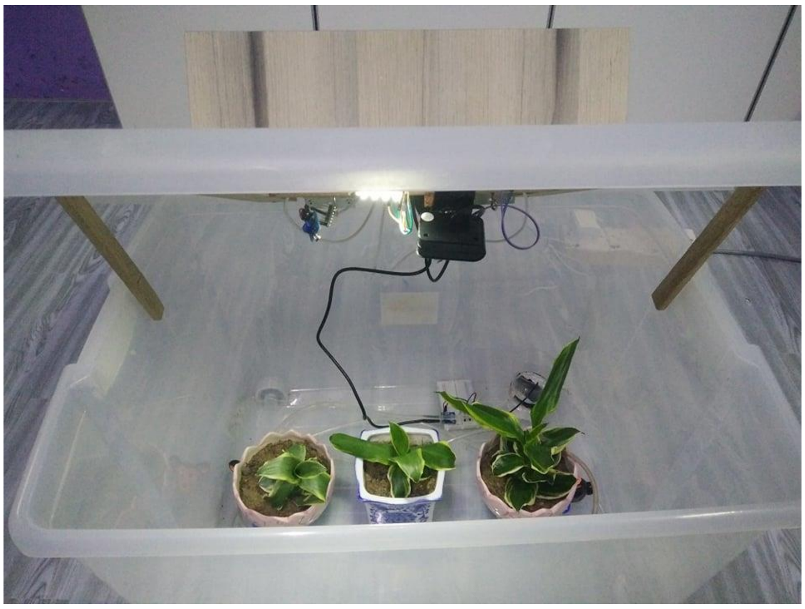


Figure 4.14 Overall Hardware Implementation (2)

4.4.8 Google Cloud Platform

There are several technologies and tools used to develop the proposed solution. The first technology used is cloud computing technology. The reason for choosing the Google Cloud platform instead of other cloud platforms is because it is easier to learn as compared to other cloud platforms. But at the same time, Google Cloud Platform also provides multiple services in order to support IoT solutions. There are a few types of Google Cloud services to be used. First, Google Cloud IoT Core is used to manage multiple IoT devices connected to the cloud and establish a secure connection in between.

Besides, Google Cloud Pub/Sub service is used to receive the message sent from the devices that registered under IoT core. Google Cloud Pub/Sub service provides asynchronous messaging between publisher and subscriber, allows data streaming from multiple sources, and also other reliability messaging features. (Google Cloud, n.d.). Besides, Google Cloud Function also acts as a fully-managed computing engine to provide data processing, action triggering and other computing features. It allows users to simply upload their code in Google Cloud Function and perform the desired tasks without any extra concern about server management. Other than that, Google Cloud Scheduler is also used to trigger Cloud Function based on the schedule set in order to monitor conditions of different sensors. For the Google Compute Engine and Google Cloud storage services, they are used to support video streaming and photo storing of Raspberry pi’s camera module feature.

4.4.9 Firebase

In this project, Firebase is used as a database to store, access, and manage data. By using Firebase Realtime Database, any updates in the database can be listened in real-time. For instance, when data added to a specific path in the database, this event can trigger Cloud Function to perform some processing task or update the interface of the mobile application. Other than database, Firebase Authentication service is used to allow account authentication and registration features by the mobile application. With the combination of Cloud services and Firebase, IoT solutions can be built, customised, visualised, and managed easily by the users. By developing a cloud-based IoT platform, users no longer need to worry about how to configure a cloud platform to support their IoT solution.

4.4.10 Android Studio

Android Studio is a development tool used in this project. It is an officially Integrated Development Environment to develop Android application (Android Developer, n.d.). Android Studio provides many features to reduce app development time such as drag and drop interface, and also great integration with the third party. The reason for choosing Android Studio instead of other android development tools is because it has built-in support for Google Cloud Platform and Firebase services such as Firebase and Google Cloud function.

4.4.11 Java Language

Java language is one of the programming language used to develop the mobile application. Besides its cross-platform ability, the main reason of choosing Java is due to its large amount of API library provided. It will save a lot of development works and also ensure flexibility of the project. Besides, the object-oriented model of Java also ensures that the code is well-structured and easy to maintain.

4.4.12 MPAndroidChart API

MPAndroidChart API is a powerful library that supports graph/chart view in Android Application. There are multiple types of graph/chart view supported including line chart, bar chart, Pie chart and other. This API in this project is used to develop a line chart for mobile application in order to let users have a view regarding past week average sensor data. The main reason for choosing this API is because it also supports interactive graph for users.

4.3.13 Google Vision API

In this project, Google Vision API is used to develop a QR code scanner to add a device via scanning QR code. It supports both 1D barcodes and 2D barcodes. The barcode detection process of Google Vision API is fast and accurate enough for this project development.

4.4.14 VLC Media Player API

The project’s mobile application also uses VLC Media Player API to provide video streaming feature to users.

4.4.15 Node.js 8

Node.js 8 will be the runtime environment chosen in Google Cloud Function. It is used to execute the Javascript Code deployed in Google Cloud Function.

4.5 Implementation Issues and Challenges

There were some problems encountered during the implementation process. The first problem faced was to ensure a secure connection between the Raspberry Pi and the application users. For instance, how to ensure that the device will not be used by unauthorised users? After a discussion with the supervisor, an approach was figured out to solve this problem. Instead of allowing everyone to connect the devices by simply scan nearby devices or scan the QR code, using extra password authentication in each device would be a good way to ensure that only authorized users can use the devices. In order to connect to a Raspberry Pi, users need to key in device password from the mobile application.

Another challenge of this project was the asynchronous behavior of the Firebase Realtime Database. This problem only arose during the project development period. It spent most of the time in debugging this issue due to the inexperienced in handling asynchronous event listeners in the programming environment. In order to handle Callback from Firebase SDK in android application, the code sequences were reordered and structured to ensure all the logics work correctly. Besides that, Aysnc and Await keywords were also used to ensure that a Callback was fulfilled before starting another part of logic that required the value return from the event listener. This approach was mainly used in the codes deployed in Cloud Function. Last but not least, there are still other implementation challenges such as user interface design, and database design. Although the problems solving process is tedious and challenging, it is a necessary process to grow from a novice to an expert in software development.

4.6 System Testing

4.6.1 Use Case Testing

Use Case	**Test Condition**	**Test Input**	**Expected Result**	**Actual Result**	**Pass/ Fail**
Register Account	Main Flow	Follow main flow steps with valid Email and password	System registers the user and redirects to Manage Device page. (MainHome activity).	System registers the user and redirects to Manage Device page. (MainHome activity).	Pass
	Alternate Flow – Invalid Email or Password	Follow alternate flow steps	System displays error message.	System displays error message.	Pass
Login Account	Main Flow	Follow main flow steps	System registers the user and redirects to Manage Device page. (MainHome activity).	System registers the user and redirects to Manage Device page. (MainHome activity).	Pass
	Alternate Flow – Invalid Email or Password	Follow alternate flow steps	System displays error message.	System displays error message.	Pass
Logout Account	Main Flow	Follow main flow steps	System logout the user and redirects to Login/Register page (MainActivity activity).	System logout the user and redirects to Login/Register page (MainActivity activity).	Pass

Manage Device	Main Flow	Follow main flow steps	Users able to view a list of added devices.	Users able to view a list of added devices.	Pass
Add Device	Main Flow	Follow main flow steps	System updates database and redirects to Manage Device page. (MainHome activity) .	System updates database and redirects to Manage Device page. (MainHome activity) .	Pass
	Alternate Flow – Services Denied	Follow alternate flow steps	System redirects to Manage Device page. (MainHome activity)	System redirects to Manage Device page. (MainHome activity)	Pass
	Alternate Flow- Scan Nearby Bluetooth Devices	Follow alternate flow steps	System updates database and redirects to Manage Device page. (MainHome activity) .	System updates database and redirects to Manage Device page. (MainHome activity) .	Pass
	Alternate Flow – Connection Failure	Follow alternate flow steps	System displays error message	System displays error message	Pass
	Alternate Flow – Invalid Device Password or Device is Added already	Follow alternate flow steps	System displays error message	System displays error message	Pass
	Alternate Flow – Have Internet Connection	Follow alternate flow steps	System updates database and redirects to Manage Device	System updates database and redirects to Manage Device	Pass

			page. (MainHome activity) .	page. (MainHome activity) .	
	Alternate Flow – No Internet Connection (Input with invalid WIFI name and WIFI password)	Follow alternate flow steps	System displays error message.	System displays error message.	Pass
Edit Device Name	Main Flow	Follow main flow steps	System updates database and updates Manage Device page (MainHome activity).	System updates database and updates Manage Device page (MainHome activity).	Pass
Remove Device	Main Flow	Follow main flow steps	System updates database and Manage Device page. (MainHome activity).	System updates database and Manage Device page. (MainHome activity).	Pass
Manage Device Component	Main Flow	Follow main flow steps	User able to view a list of sensor and actuator of an added device.	User able to view a list of sensor and actuator of an added device.	Pass
Add Device Component	Main Flow	Follow main flow steps	System updates Manage Device Component page (Dashboard activity).	System updates Manage Device Component page (Dashboard activity).	Pass
	Alternate Flow – Plug In New/Existing Actuator	Follow alternate flow steps	System updates Manage Device Component page (Dashboard activity).	System updates Manage Device Component page (Dashboard activity).	Pass

	Alternate Flow – Plug In Existing Sensor	Follow alternate flow steps	System updates Manage Device Component page (Dashboard activity).	System updates Manage Device Component page (Dashboard activity).	Pass
	Alternate Flow – Connect to Previous Sensor Data (At least one related sensor ID)	Follow alternate flow steps	System updates database and Manage Device Component page (Dashboard activity).	System updates database and Manage Device Component page (Dashboard activity).	Pass
	Alternate Flow – Connect to Previous Sensor Data (No related sensor ID)	Follow alternate flow steps	System updates Manage Device Component page (Dashboard activity).	System updates Manage Device Component page (Dashboard activity).	Pass
Edit Device Component Name	Main Flow	Follow main flow steps	System updates database and Manage Device Component page (Dashboard activity).	System updates database and Manage Device Component page (Dashboard activity).	Pass
Remove Device Component	Main Flow	Follow main flow steps	System updates database and Manage Device Component page (Dashboard activity).	System updates database and Manage Device Component page (Dashboard activity).	Pass
Control Device Component	Main Flow	Follow main flow steps	System updates database and Manage Device Component page	System updates database and Manage Device Component page	Pass

			(Dashboard activity).	(Dashboard activity).	
View Live Streaming Video	Main Flow	Follow main flow steps	System plays video from cloud.	System plays video from cloud.	Pass
View Analytic Graph	Main Flow	Follow main flow steps	System displays analytic graph.	System displays analytic graph.	Pass
Manage Trigger	Main Flow	Follow main flow steps	User able to view a list of added trigger.	User able to view a list of added trigger.	Pass
Add Trigger	Main Flow	Follow main flow steps	System updates database and Trigger page (Trigger activity).	System updates database and Trigger page (Trigger activity).	Pass
Edit Trigger	Main Flow	Follow main flow steps	System updates database and Trigger page (Trigger activity).	System updates database and Trigger page (Trigger activity).	Pass
Remove Trigger	Main Flow	Follow main flow steps	System updates database and Trigger page (Trigger activity).	System updates database and Trigger page (Trigger activity).	Pass
Edit Trigger Name	Main Flow	Follow main flow steps	System updates database and Trigger page (Trigger activity).	System updates database and Trigger page (Trigger activity).	Pass
View Trigger History	Main Flow	Follow main flow steps	System displays a list of trigger history.	System displays a list of trigger history.	Pass

Table 4.1 Use Case Testing

CHAPTER 5: SYSTEM OUTCOME AND DISCUSSION

5.1 Cloud Architecture

Figure 5.1 is the final Google Cloud architecture diagram of system. All of the core features as mentioned in Chapter 3 System Design are implemented.

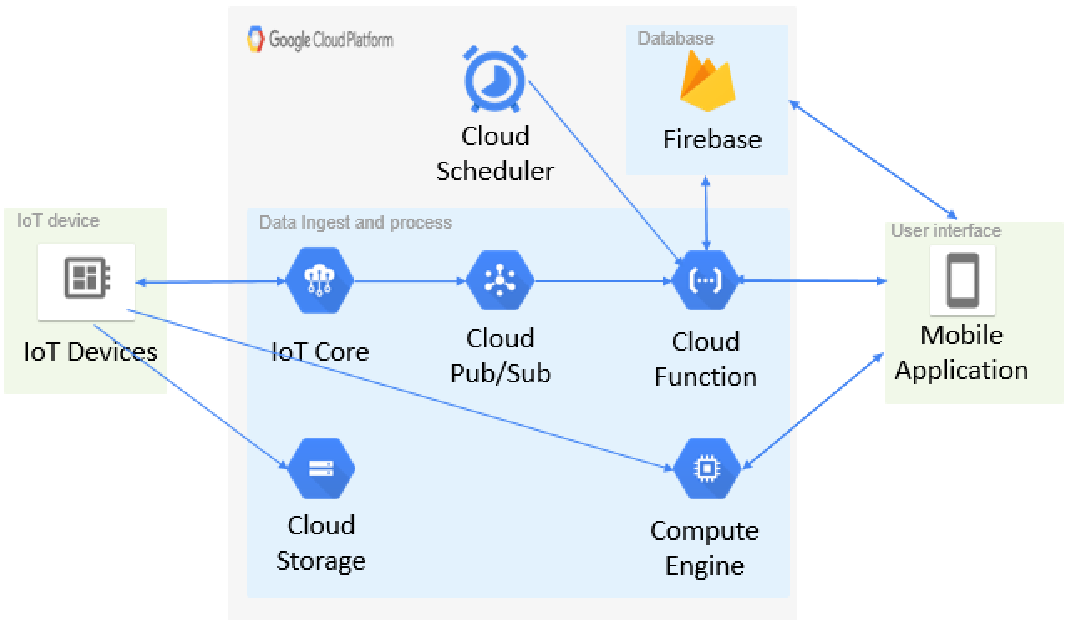
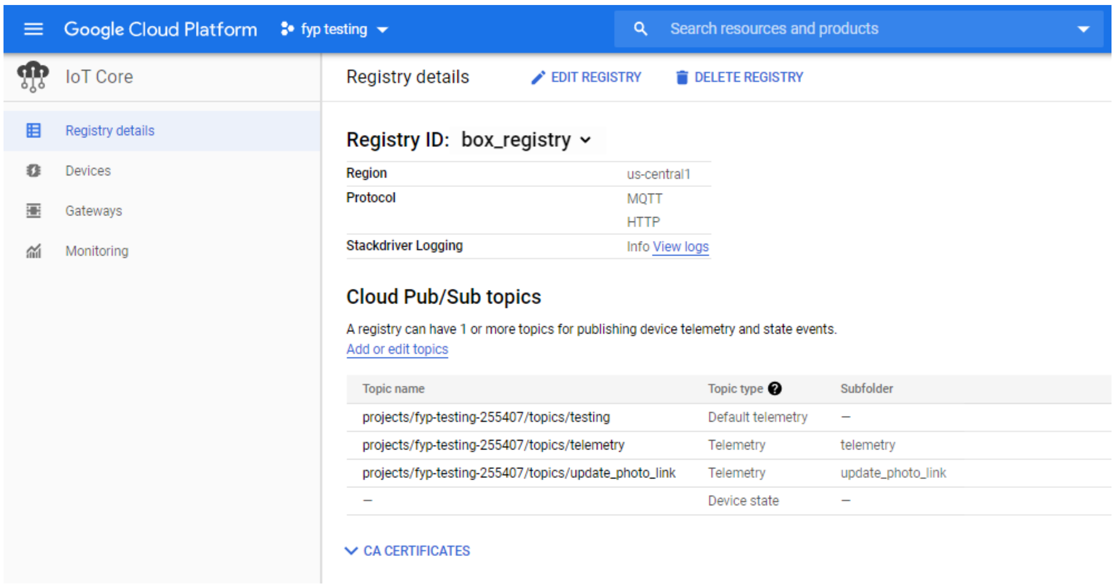


Figure 5.1 Architecture Diagram of System Deliverable

There is one registry created in IoT core to act as a gateway for devices connection. Although the registry created supports both HTTP and MQTT protocol, in this project, MQTT will be the main protocol selected for device connection due to various reasons. First, the MQTT protocol is much more lightweight than HTTP as the message header is shorter than HTTP. MQTT is more suitable for IoT devices as the battery and bandwidth usage are the concerns of IoT. Another reason is the Stackdriver logging feature provided by Google Cloud which supports the MQTT connection. This feature will capture devices error such as disconnection and publish error messages to a pub sub topic for further processing. It is extremely important in the case that the client will get an immediate update of device status when unexpected disconnection occurs. While one registry ID can register for multiple devices, the number of the registry can be scalable in the future when the demand for IoT devices increases to support different user requirements. Last but not least, IoT core also allows users to monitor the IoT environment as shown in figure 5.3.



Registry details

Registry ID: box_registry

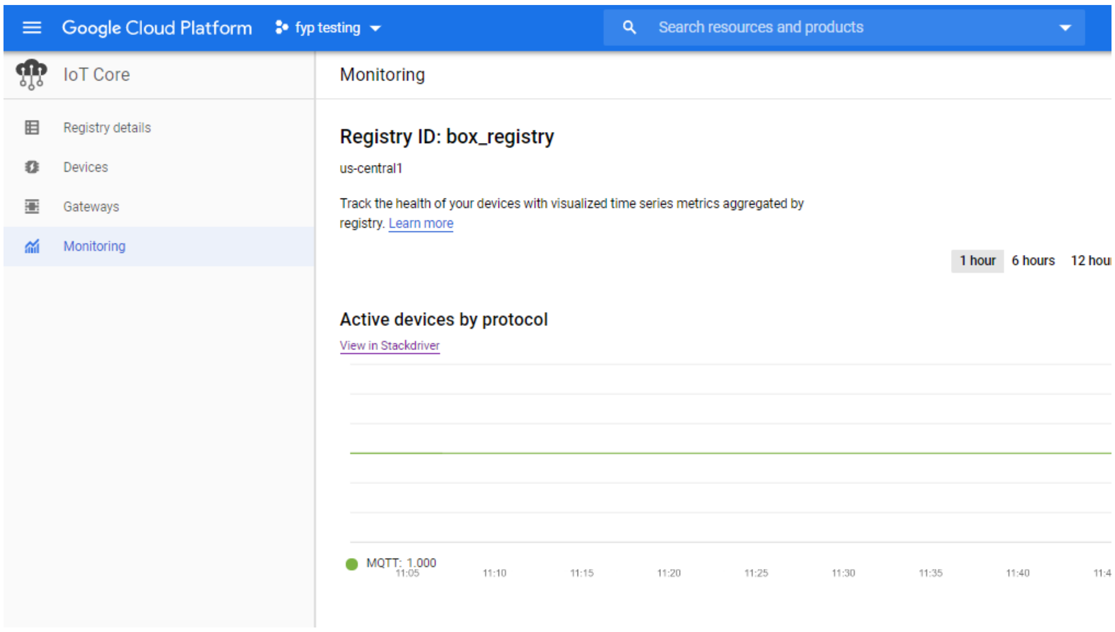
Region	us-central1
Protocol	MQTT HTTP
Stackdriver Logging	Info View logs

Cloud Pub/Sub topics

A registry can have 1 or more topics for publishing device telemetry and state events.
Add or edit topics

Topic name	Topic type	Subfolder
projects/fyp-testing-255407/topics/testing	Default telemetry	–
projects/fyp-testing-255407/topics/telemetry	Telemetry	telemetry
projects/fyp-testing-255407/topics/update_photo_link	Telemetry	update_photo_link
–	Device state	–

Figure 5.2 IoT Core Registry Details



Monitoring

Registry ID: box_registry

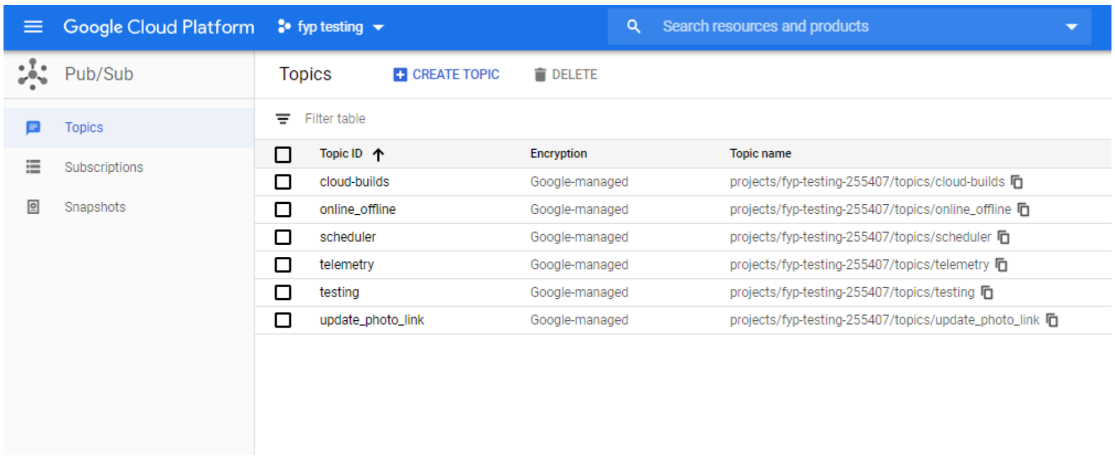
us-central1

Track the health of your devices with visualized time series metrics aggregated by registry. Learn more

Active devices by protocol

Figure 5.3 Monitoring Devices' Conditions in IoT Core

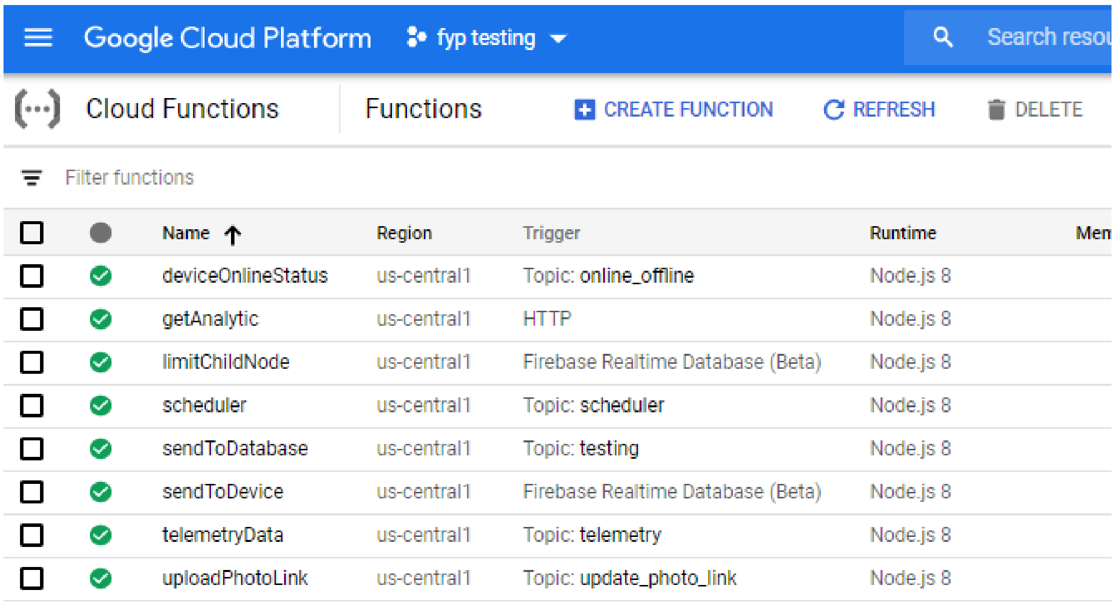
There are few Pub/Sub topics created in Cloud Pub/Sub which receive different messages from devices connected to IoT core. First, online_offline topic will receive a message when the device is connected or disconnected from the cloud. By processing this message in cloud function, any unexpected disconnection from the devices can be monitored. The second topic - telemetry will receive all the telemetry data from the devices' published messages. The third topic – testing is the first topic created in this project. It is the default topic to receive the devices' published message. In the current case, this topic is used to receive all related devices data such as device status and things connected except for telemetry data. Scheduler topic is a topic created by Cloud Scheduler to assist in automation feature of this system. The last topic created is update_photo_link which will receive captured photo storage link from Raspberry Pi and it is used to trigger a cloud function to store the link in Firebase Realtime Database.



Topic ID	Encryption	Topic name
cloud-builds	Google-managed	projects/fyp-testing-255407/topics/cloud-builds
online_offline	Google-managed	projects/fyp-testing-255407/topics/online_offline
scheduler	Google-managed	projects/fyp-testing-255407/topics/scheduler
telemetry	Google-managed	projects/fyp-testing-255407/topics/telemetry
testing	Google-managed	projects/fyp-testing-255407/topics/testing
update_photo_link	Google-managed	projects/fyp-testing-255407/topics/update_photo_link

Figure 5.4 Pub/Sub Topics Created in Cloud Pub/Sub

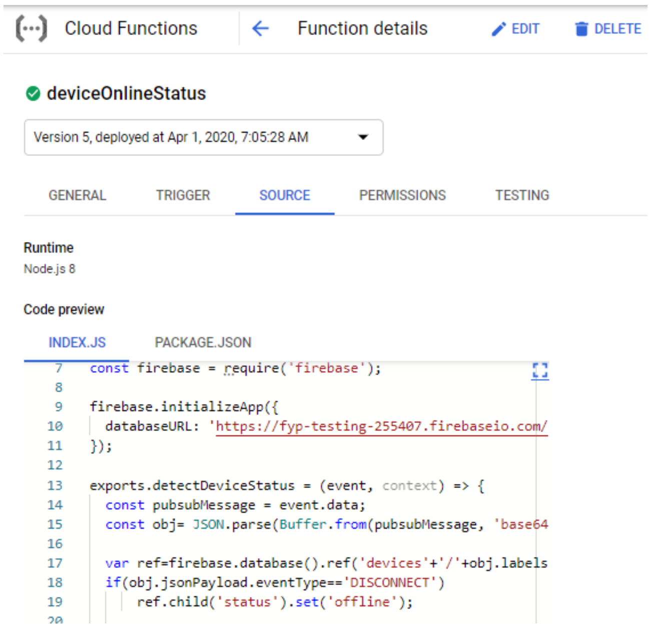
There are 8 cloud functions deployed in Google Cloud Function to perform different tasks.



	Name	Region	Trigger	Runtime	Mem
✓	deviceOnlineStatus	us-central1	Topic: online_offline	Node.js 8	
✓	getAnalytic	us-central1	HTTP	Node.js 8	
✓	limitChildNode	us-central1	Firebase Realtime Database (Beta)	Node.js 8	
✓	scheduler	us-central1	Topic: scheduler	Node.js 8	
✓	sendToDatabase	us-central1	Topic: testing	Node.js 8	
✓	sendToDevice	us-central1	Firebase Realtime Database (Beta)	Node.js 8	
✓	telemetryData	us-central1	Topic: telemetry	Node.js 8	
✓	uploadPhotoLink	us-central1	Topic: update_photo_link	Node.js 8	

Figure 5.5 Cloud Functions

First, deviceOnlineStatus will be triggered every time when the online_offline topic receives a message. By processing the message, device status can be tracked and update to the Firebase Realtime Database immediately when receive the disconnection message.



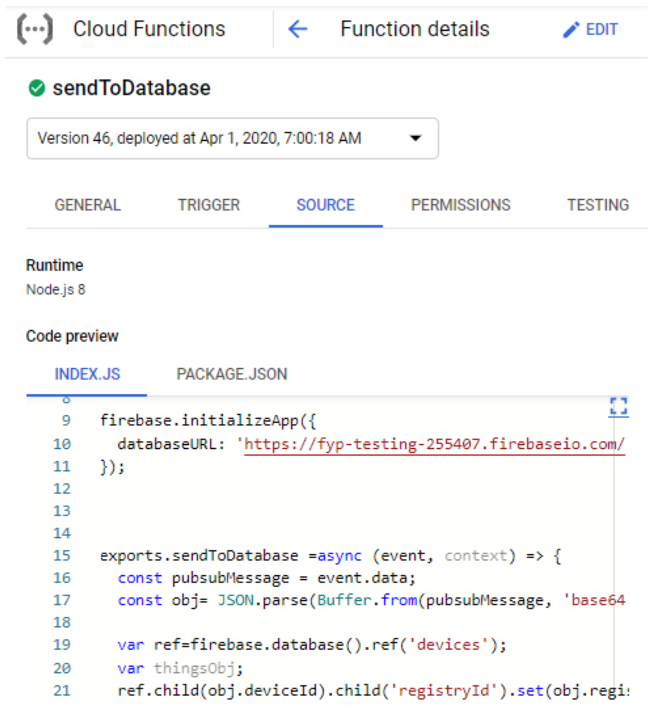
Runtime
Node.js 8

Code preview

```
const firebase = require('firebase');

firebase.initializeApp({
  databaseURL: 'https://fyp-testing-255407.firebaseio.com/
});

exports.detectDeviceStatus = (event, context) => {
  const pubsubMessage = event.data;
  const obj= JSON.parse(Buffer.from(pubsubMessage, 'base64

  var ref=firebase.database().ref('devices'+'/'+obj.labels
  if(obj.jsonPayload.eventType=='DISCONNECT')
      ref.child('status').set('offline');
```

Figure 5.6 DeviceOnlineStatus Cloud Function

The second function – sendToDatabase is used to update all devices' information except telemetry data to the Firebase Realtime Database. For instance, the current status of the device, the number of things connected to the devices, the details of things and other related device information. This function will be triggered when there is a message published to default topic – testing. There are several conditions that a message will be published to the default topic. First of all, the devices' status and all the connected things details will be sent as a message to the topic after turning on or before turning off the device. Besides, whenever users plug in or plug out a sensor or actuator from the devices when the devices are online, a message regarding the sensor or actuator information will be sent to the topic in order to update the database in real-time.



sendToDatabase

Version 46, deployed at Apr 1, 2020, 7:00:18 AM

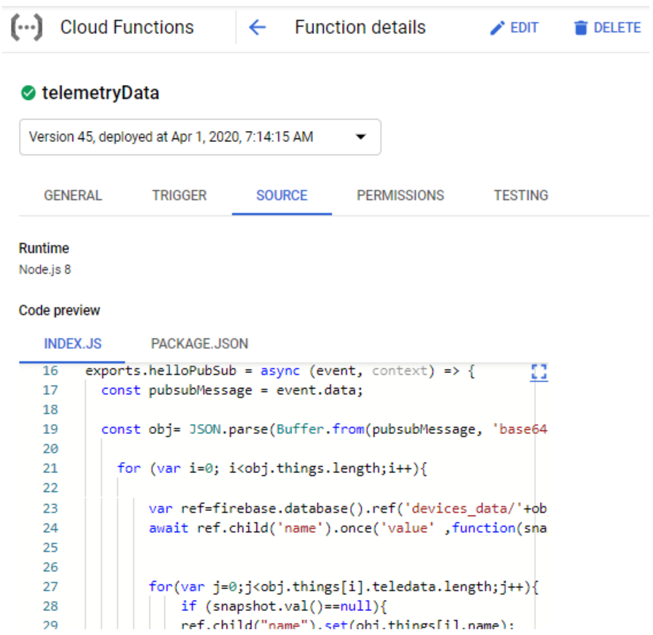
Runtime
Node.js 8

Code preview

```
firebase.initializeApp({
  databaseURL: 'https://fyp-testing-255407.firebaseio.com/
});



exports.sendToDatabase =async (event, context) => {
  const pubsubMessage = event.data;
  const obj= JSON.parse(Buffer.from(pubsubMessage, 'base64

  var ref=firebase.database().ref('devices');
  var thingsObj;
  ref.child(obj.deviceId).child('registryId').set(obj.regi
```

Figure 5.7 SendToDatabase Cloud Function

The third function telemetryData will be triggered once the telemetry topic receives a message from devices. The telemetry data includes timestamp, value, data type, device Id, and thing Id.



telemetryData

Version 45, deployed at Apr 1, 2020, 7:14:15 AM

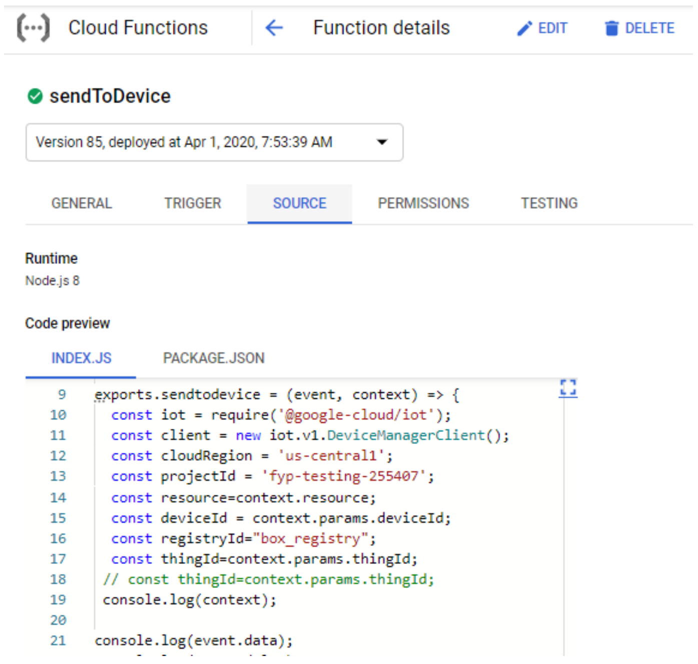
Runtime
Node.js 8

Code preview

```
exports.helloPubSub = async (event, context) => {
  const pubsubMessage = event.data;

  const obj= JSON.parse(Buffer.from(pubsubMessage, 'base64

    for (var i=0; i<obj.things.length;i++){

        var ref=firebase.database().ref('devices_data/'+ob
        await ref.child('name').once('value' ,function(sna


        for(var j=0;j<obj.things[i].teledata.length;j++){
            if (snapshot.val()==null){
            ref.child("name").set(obj.things[i].name);
```

Figure 5.8 TelemetryData Cloud Function

Another cloud function created is sendToDevice which plays as a role to send commands back to the devices from the mobile application. This function allows application users to switch on/off actuator from the user interface provided. Once there is a change detected in the Firebase Realtime Database, this function will be triggered to send the command payload to the devices for controlling purpose.



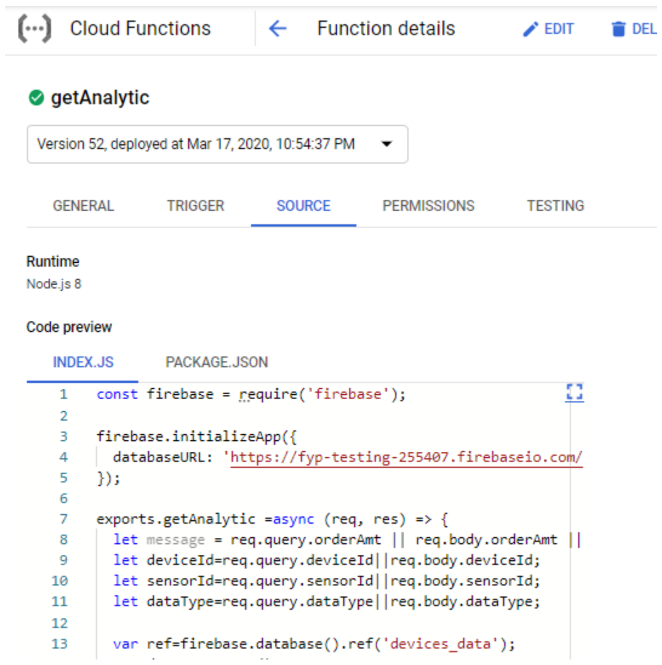
Runtime
Node.js 8

Code preview

```
exports.sendtodevice = (event, context) => {
  const iot = require('@google-cloud/iot');
  const client = new iot.v1.DeviceManagerClient();
  const cloudRegion = 'us-central1';
  const projectId = 'fyp-testing-255407';
  const resource=context.resource;
  const deviceId = context.params.deviceId;
  const registryId="box_registry";
  const thingId=context.params.thingId;
 // const thingId=context.params.thingId;
 console.log(context);

console.log(event.data);
```

Figure 5.9 SendToDevice Cloud Function

GetAnalytic Cloud Function is created to perform calculation task of average sensor data per day in past week. Instead of performing calculation task in the mobile application, Cloud Function is a better option to reduce the burden of mobile phone processing power. It is triggered when an HTTP request comes from the mobile application and the calculation result will be a dataset to generate a line chart in the mobile application.



getAnalytic

Version 52, deployed at Mar 17, 2020, 10:54:37 PM

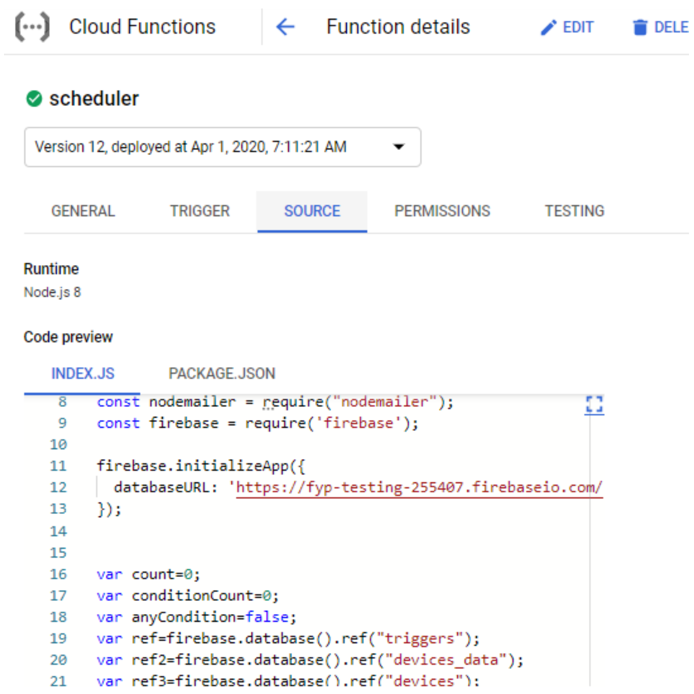
Runtime
Node.js 8

Code preview

```
const firebase = require('firebase');

firebase.initializeApp({
  databaseURL: 'https://fyp-testing-255407.firebaseio.com/
});

exports.getAnalytic =async (req, res) => {
  let message = req.query.orderAmt || req.body.orderAmt ||
  let deviceId=req.query.deviceId||req.body.deviceId;
  let sensorId=req.query.sensorId||req.body.sensorId;
  let dataType=req.query.dataType||req.body.dataType;

  var ref=firebase.database().ref('devices_data');
```

Figure 5.10 GetAnalytic Cloud Function

Scheduler is another Cloud Function that subscribed to scheduler topic in Pub/Sub. This function will be triggered every 5min in this project to monitor triggers set by users. In the code written, every triggers in the database will be checked every 5 min to detect fulfilled conditions and perform corresponding actions that set by users. This Cloud Function is working like a centralised monitoring of users' IoT solutions.



scheduler

Version 12, deployed at Apr 1, 2020, 7:11:21 AM

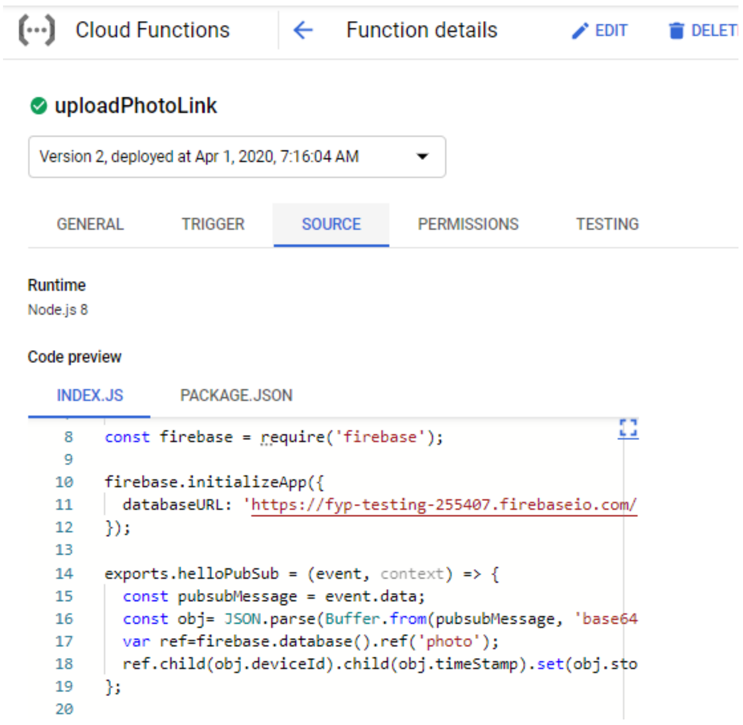
Runtime
Node.js 8

Code preview

```
const nodemailer = require("nodemailer");
const firebase = require('firebase');

firebase.initializeApp({
  databaseURL: 'https://fyp-testing-255407.firebaseio.com/
});


var count=0;
var conditionCount=0;
var anyCondition=false;
var ref=firebase.database().ref("triggers");
var ref2=firebase.database().ref("devices_data");
var ref3=firebase.database().ref("devices");
```

Figure 5.11 Scheduler Cloud Function

UploadPhotoLink Cloud Function is a function that subscribes to Update_Photo_Link topic. It is triggered when a message is sent to the topic and store the photo link information in Firebase Realtime database for future development purpose.



Version 2, deployed at Apr 1, 2020, 7:16:04 AM

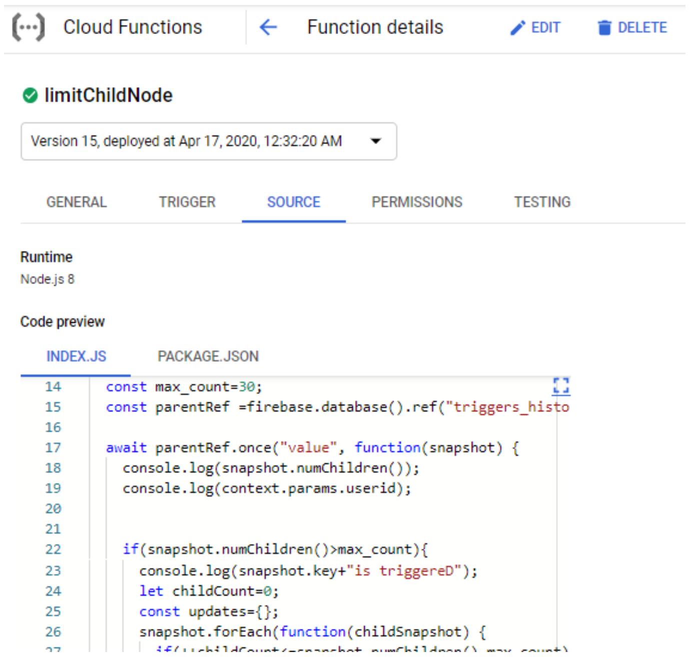
Runtime
Node.js 8

Code preview

```
const firebase = require('firebase');

firebase.initializeApp({
  databaseURL: 'https://fyp-testing-255407.firebaseio.com/
});

exports.helloPubSub = (event, context) => {
  const pubsubMessage = event.data;
  const obj= JSON.parse(Buffer.from(pubsubMessage, 'base64
  var ref=firebase.database().ref('photo');
  ref.child(obj.deviceId).child(obj.timeStamp).set(obj.sto
};

```

Figure 5.12 UploadPhotoLink Cloud Function

To prevent the database from storing a large amount of unnecessary information, LimitChildNode function is created to limit the number of nodes stored in the database. This function is fired when a trigger history message is added in the trigger history path to check the number of nodes in the path. If the number of nodes exceeds the maximum amount set in code written in the function, the oldest node will be deleted to ensure that only 30 records of trigger history for a user.



Runtime
Node.js 8

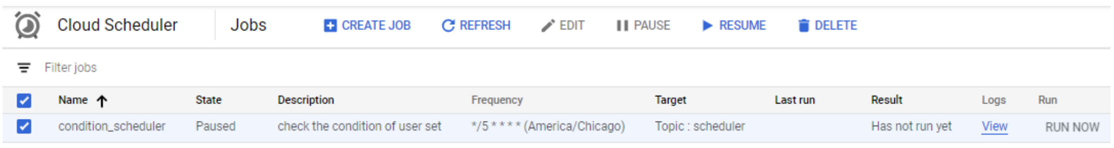
Code preview

INDEX.JS PACKAGE.JSON

```
const max_count=30;
const parentRef =firebase.database().ref("triggers_histo

await parentRef.once("value", function(snapshot) {
  console.log(snapshot.numChildren());
  console.log(context.params.userid);


  if(snapshot.numChildren()>max_count){
    console.log(snapshot.key+"is triggereD");
    let childCount=0;
    const updates={};
    snapshot.forEach(function(childSnapshot) {
```

Figure 5.13 LimitChildNode Cloud Function

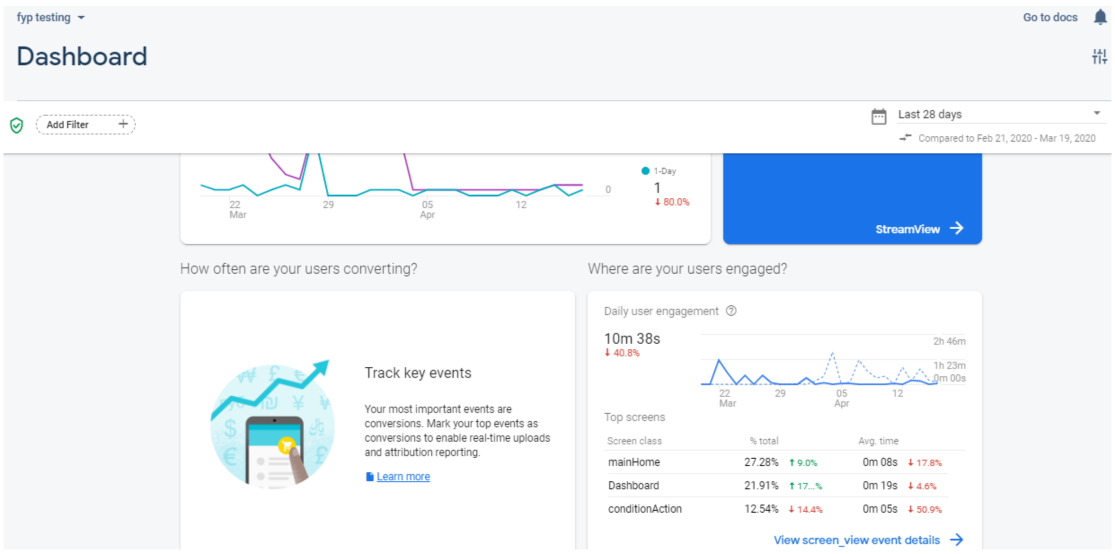
Other than Cloud Function, Cloud Scheduler is used to trigger certain Cloud Functions based on schedule sets. In this project, a scheduler task is created to trigger the Scheduler Cloud Function every 5 min to monitor automation rules set by users.



✓	Name	State	Description	Frequency	Target	Last run	Result	Logs	Run
✓	condition_scheduler	Paused	check the condition of user set	*/5 * * * * (America/Chicago)	Topic : scheduler		Has not run yet	View	RUN NOW

Figure 5.14 Cloud Scheduler

Firebase is one of the core services used in this project. It provides a lot of features such as authentication and real-time database to integrate easily with the Android application and other Google Cloud services. By implementing the Firebase in the Android application, the overall status of the application can be monitored easily through the Firebase Analytic Dashboard. As shown in figure 5.15, the interaction between the users and application can be visualized from the dashboard provided.

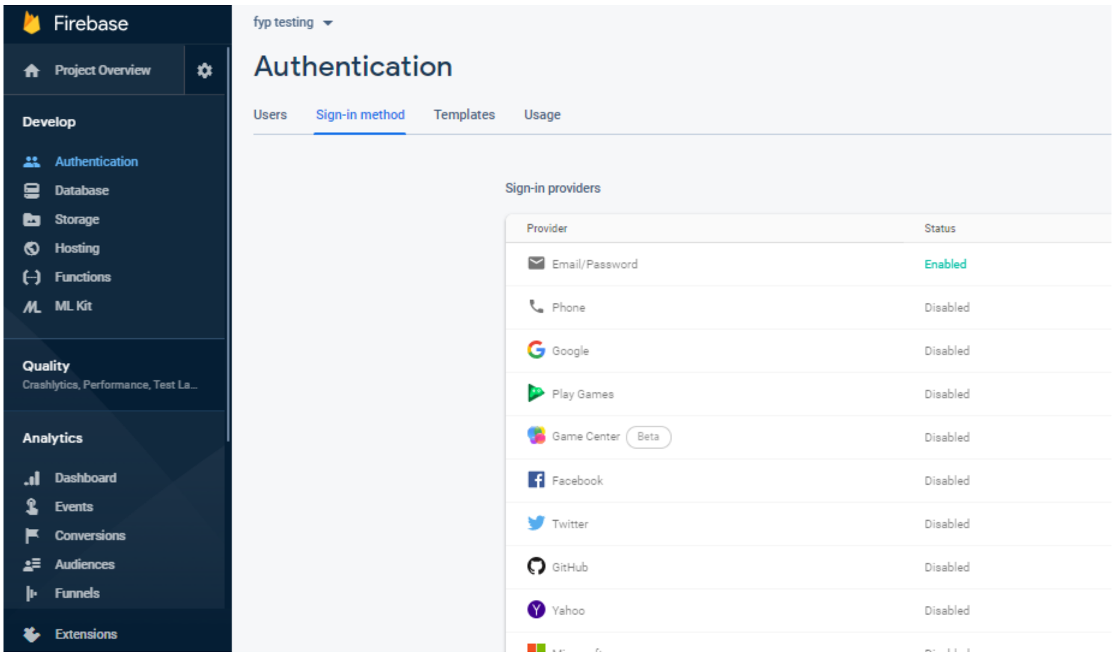

fyp testing
Dashboard
Go to docs
Add Filter
Last 28 days
Compared to Feb 21, 2020 - Mar 19, 2020
1-Day
StreamView
How often are your users converting?
Track key events
Your most important events are conversions. Mark your top events as conversions to enable real-time uploads and attribution reporting.
Learn more
Where are your users engaged?
Daily user engagement
10m 38s
40.8%
View screen_view event details

Top screens

Screen class	% total		Avg. time	
mainHome	27.28%	↑ 9.0%	0m 08s	↓ 17.8%
Dashboard	21.91%	↑ 17...%	0m 19s	↓ 4.6%
conditionAction	12.54%	↓ 14.4%	0m 05s	↓ 50.9%

Figure 5.15 Firebase Analytic Dashboard

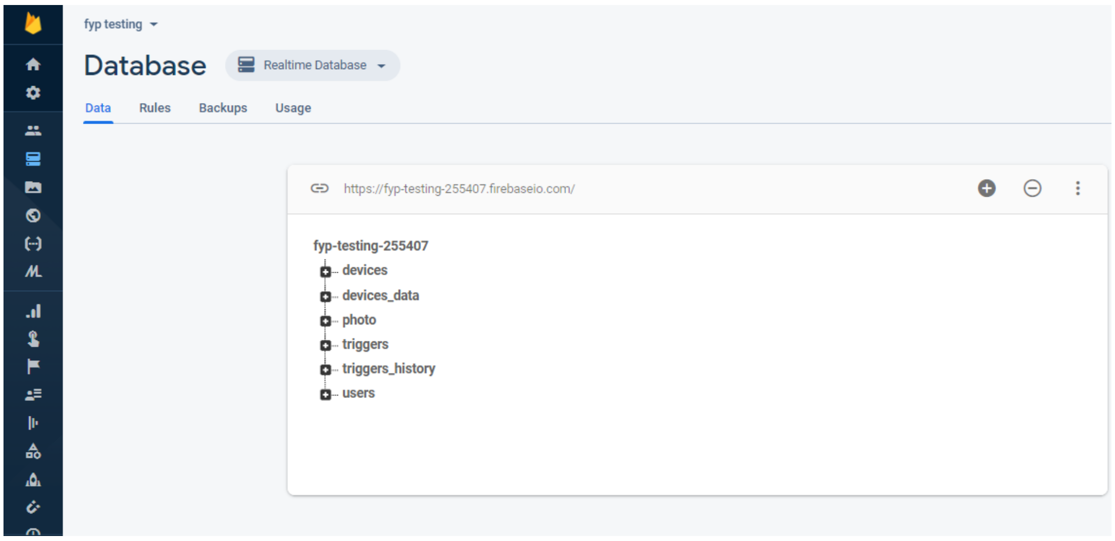
In this project, Firebase Authentication SDK is used in application development. By using the SDK, application development time is greatly reduced to implement user authentication feature in the application. Currently, the authentication method using in the project is email and password authentication. There are multiple authentication methods that can be implemented in the future from Firebase Authentication such as Google, Facebook, and Twitter.



Provider	Status
Email/Password	Enabled
Phone	Disabled
Google	Disabled
Play Games	Disabled
Game Center Beta	Disabled
Facebook	Disabled
Twitter	Disabled
GitHub	Disabled
Yahoo	Disabled

Figure 5.16 Firebase Authentication

Other than authentication, Firebase Realtime Database is used in this project to act as a centralized database system. The reason for choosing this Database instead of others is due to the NoSQL database structure which is more suitable for IoT development. Besides that, by using Firebase active event listener in Android application, updates in the database can be reflected in the application immediately. For instance, real-time sensor data update in the user interface, dynamic control panel, and other interfaces update in the mobile application.



fyp-testing-255407
- devices
- devices_data
- photo
- triggers
- triggers_history
- users

Figure 5.17 Firebase Realtime Database

5.2 Android Mobile Application

Besides back end services from Google Cloud and Firebase, front end application which interact with the end users is also playing a vital role in software development. In this project, an Android mobile application is developed to fulfil the second project objective – provide a user-friendly IoT monitoring interface for the proposed platform in order to reduce the learning gap of customising an IoT solution.

Figure 5.18 shows the Login/Register activity in the application. All users will be navigated to this activity once they open the mobile application. Users can register a new account or login an existing account to the cloud platform. After login their accounts, this activity will be only shown when users logout their accounts explicitly. Currently, it only supports email and password authentication. In the future, more types of login methods may be implemented such as Facebook, Google and Twitter.

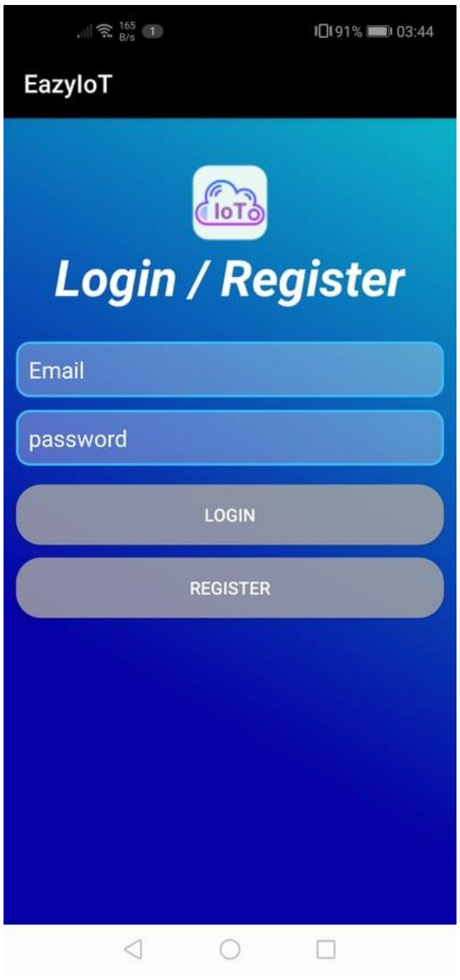


Figure 5.18 Login/Register Activity

Figure 5.19, figure 5.20 and figure 5.21 show the home page of the applications. Users will be navigated to this activity once the users logged in to their accounts. Figure 5.19 is the interface that shows to users when no device is added. If users tend to add a device, they only need to go through a simple device adding process by pressing “+” button in the home page. After adding devices, users can view all the devices added as shown in figure 20. The centralized monitoring and controlling feature is one of the main contributions of this project which allows users to control and monitor their devices through a single application interface. Some of the existing products can only support connections up to one device. Other than that, users can edit the devices’ details from the context menu by long pressing a device as shown in figure 5.21. From the context menu, users can choose to edit the name of the devices or delete the device added.

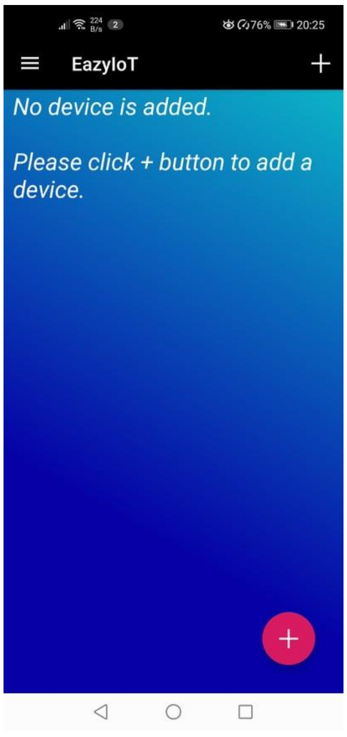


Figure 5.19 MainHome Activity (No Device)

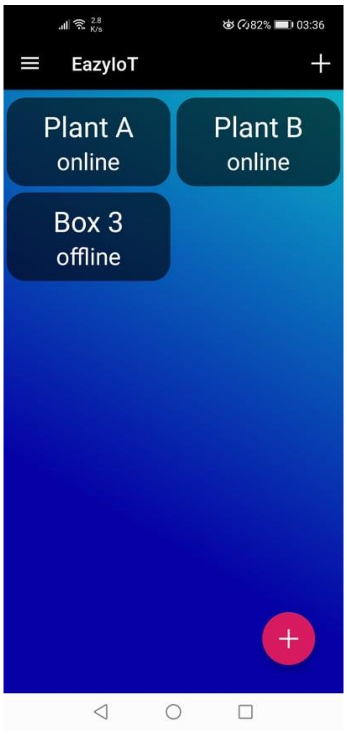


Figure 5.20 MainHome Activity (Device Added)

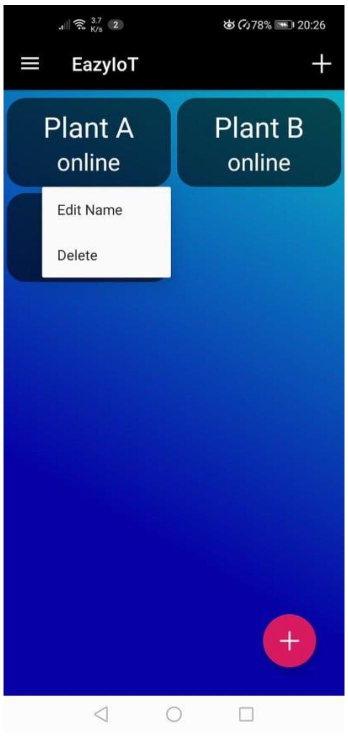


Figure 5.21 MainHome Activity (Context Menu)

When users press “+” button on the top right side of the home page or floating button, or “+Add device” button in the drawer menu, users will be navigated to Add Device activity. In this activity, users can register devices by scanning QR code or nearby Bluetooth devices. Scanning QR code attached on the device will be easy for users to add devices without too much technical knowledge required. Users only need to type in correct device password for authentication purpose. The device authentication process is one of the novelties in this project to ensure that no unauthorized users can use others’ devices. For instance, even an unauthorized user scans the QR code attached on the devices, the user is not able to register the devices without entering the correct password. In addition, WIFI setting inputs will be only needed when the device does not have an Internet connection in order to set an Internet connection. The simple device onboarding process will greatly reduce the effort of users in device installation process.

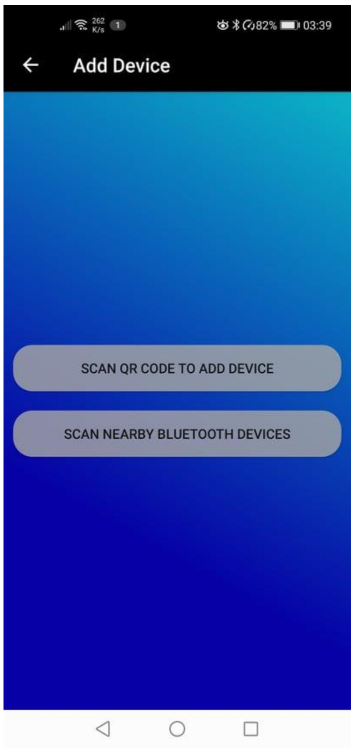
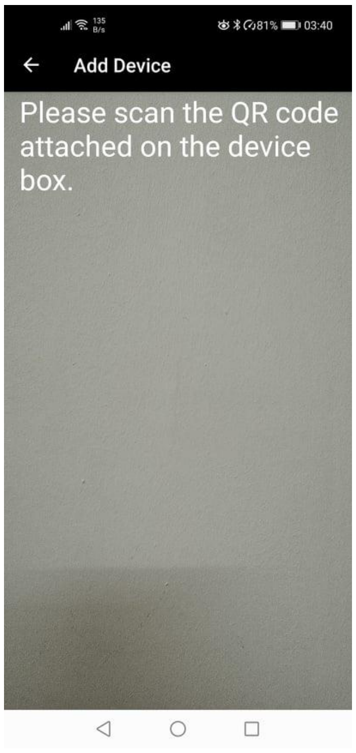


Figure 5.22 AddDeviceMethod Activity



Please scan the QR code attached on the device box.

Figure 5.23 Scan QR Code to Add Device

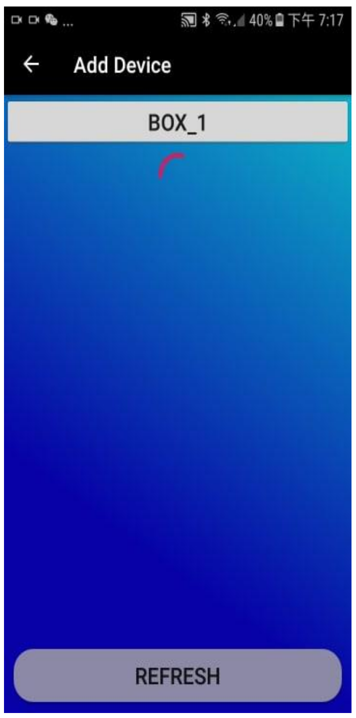
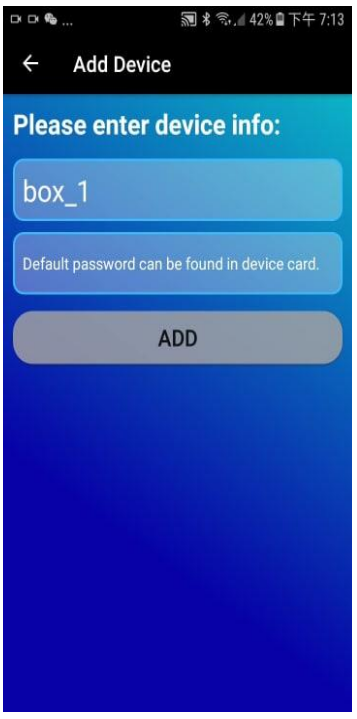


Figure 5.24 Scan Nearby Bluetooth Device to Add Device

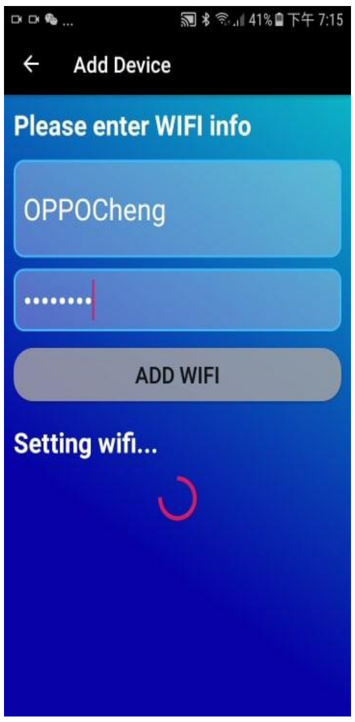


Please enter device info:

box_1

Default password can be found in device card.

Figure 5.25 Device Authentication



Please enter WIFI info

OPPOCheng

••••••••

Setting wifi...

Figure 5.26 WIFI Installation

After registration of devices, users can press am online status device in home page. Then, users will be navigated to another detailed dashboard of the device component as shown in figure 5.27. In this activity, users can visualize all the components connected to the device in real-time. For example, updates in sensor data will reflect in the interface immediately to ensure a better monitoring purpose. Besides monitoring the sensor data, the actuators can be controlled by users from the interface. Besides, users can also edit a sensor/actuator's name from the context menu by long pressing the sensor/actuator in the dashboard.

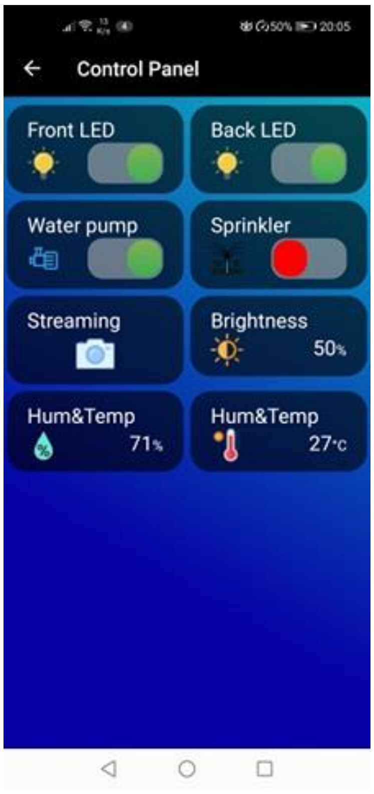


Figure 5.27 Dashboard Activity

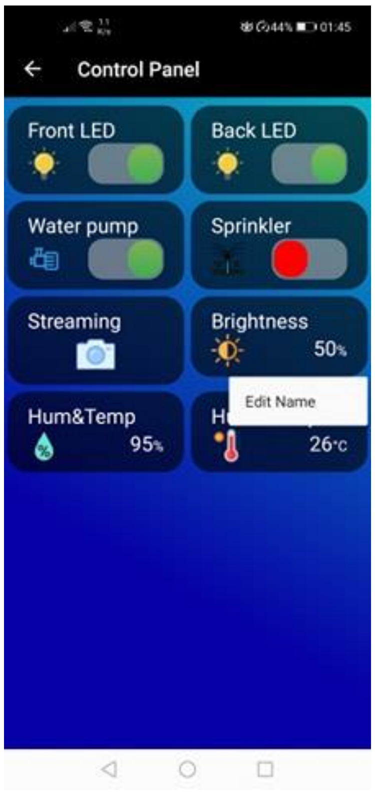
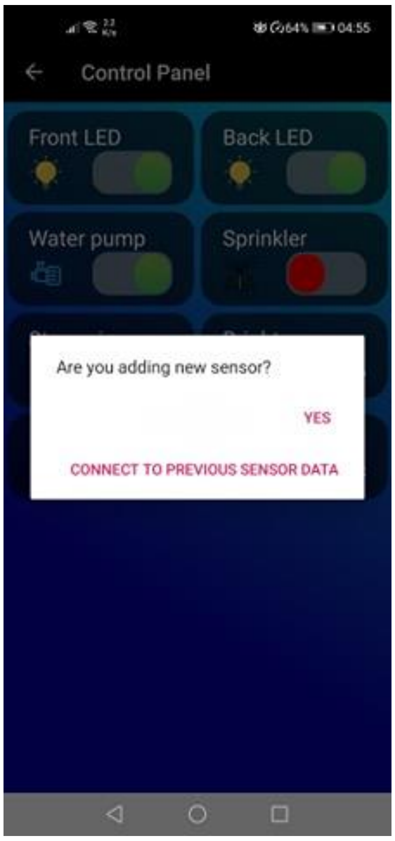


Figure 5.28 Dashboard Activity (Context Menu)

Moreover, users can enjoy a plug and play feature from the device component interface. All the actions of plug in and plug out actuator/sensor from the Raspberry Pi will update the interface immediately. Only plugged in actuator/sensor will be shown in this interface. Furthermore, the system is smart enough to determine that whether the new plugged in sensor is existing in the database. If yes, the interface will update immediately. Otherwise, users can select either add this sensor as a new thing or connect to the existing data from the dialog box as shown in figure 5.29 and figure 5.30.

This approach is used to ensure the sensor data continuity for the users. For instance, if the users lost a particular sensor, the system allows the users to copy the data from the lost sensor to the new added sensor.

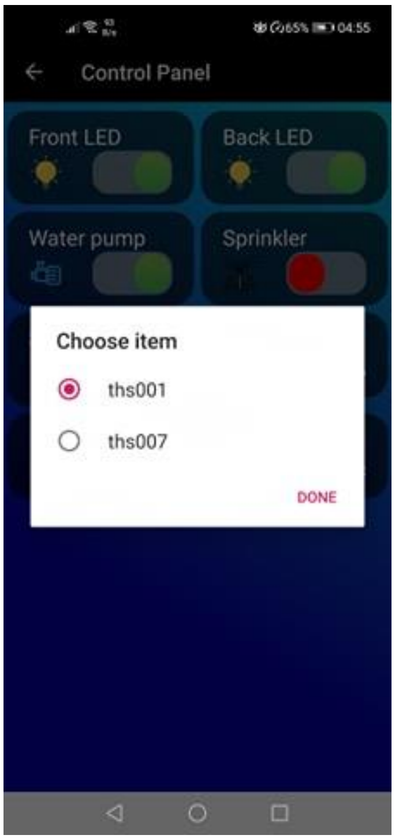


Are you adding new sensor?

YES

CONNECT TO PREVIOUS SENSOR DATA

Figure 5.29 Dialog Box (1)



Choose item

● ths001

○ ths007

DONE

Figure 5.30 Dialog Box (2)

In order to enhance the monitoring feature, an analytic graph feature is also available in this application. Users can view each sensor analytic graph by simply pressing a sensor panel from the component dashboard of the device, then a graph will be generated according to the sensor selected. Currently, average data per day will be calculated for seven days and generate a line chart for the dataset. It allows users to gain a view on past week sensor record in a graphical view. In the future, more customisation features of the graph will be developed such as customizing graph type in order to derive insight of data for the IoT users.

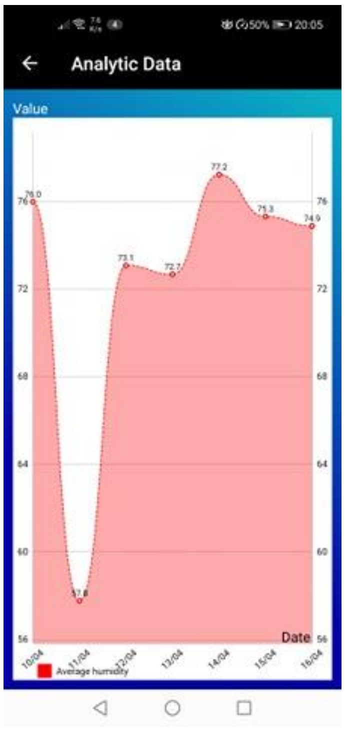
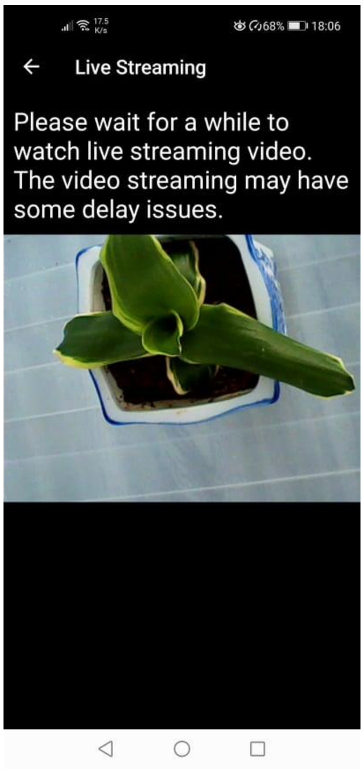


Figure 5.31 Chart Activity

If users plug in a camera sensor into the Raspberry Pi, a camera panel will be displayed in component dashboard. It allows users to view live streaming video from the camera by pressing the camera panel.



Please wait for a while to watch live streaming video. The video streaming may have some delay issues.

Figure 5.32 VideoStreaming Activity

Trigger feature is another feature highlighted in this system. It allows users to customise their preferable IoT solution by setting simple automation rule or notification. By selecting Trigger in the drawer menu as shown in figure 5.33, users will be navigated to Trigger activity. This interface will show all the created triggers by the user. By pressing "+" floating button, users will be navigated to ConditionAction activity. In this activity, users can add multiple conditions and actions in a single trigger by pressing Add a Condition or Add an Action in the interface. Besides that, users can select AND operator or OR operator to check the conditions set. Currently, users can only check the latest sensor data condition by selecting from 5 comparison operators: Greater, Equal, Smaller, Greater or Equal, Smaller or Equal. On the other hand, there are two types of actions supported – automation to turn on/off actuator and send a notification to the user's email if the condition reached.

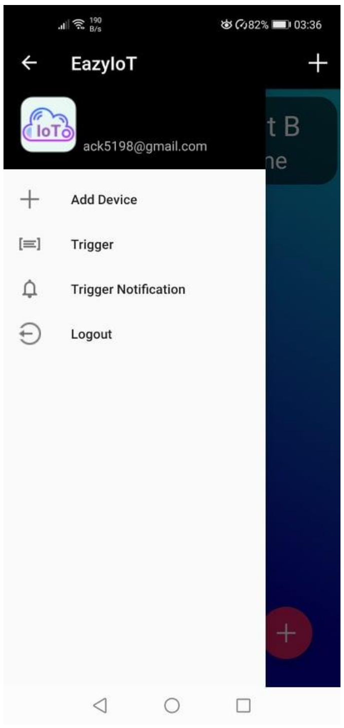
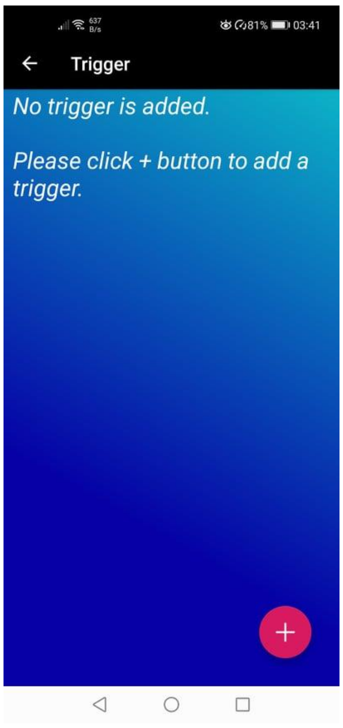


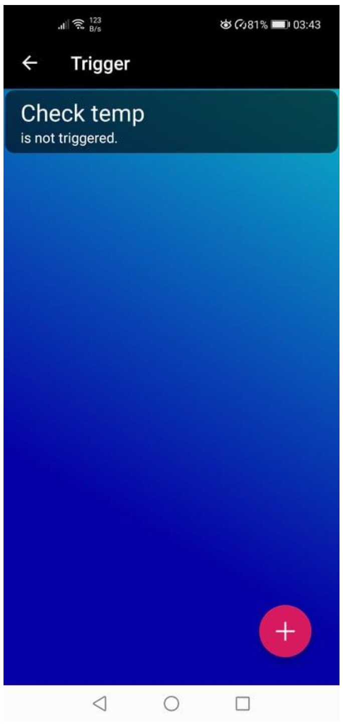
Figure 5.33 Drawer Menu



No trigger is added.

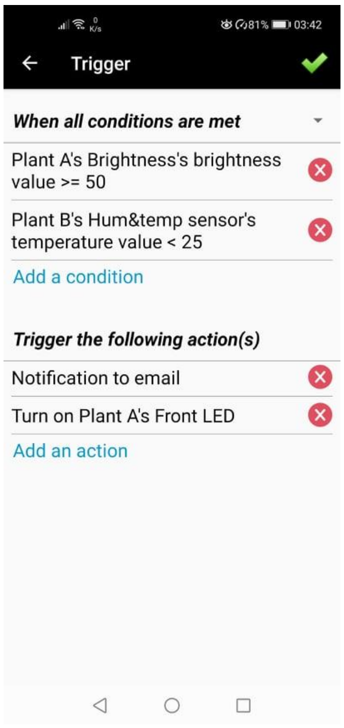
Please click + button to add a trigger.

Figure 5.34 Trigger Activity (No Trigger)



Check temp
is not triggered.

Figure 5.35 Trigger Activity (Trigger Added)



Trigger

When all conditions are met

Plant A's Brightness's brightness value >= 50

Plant B's Hum&temp sensor's temperature value < 25

Add a condition

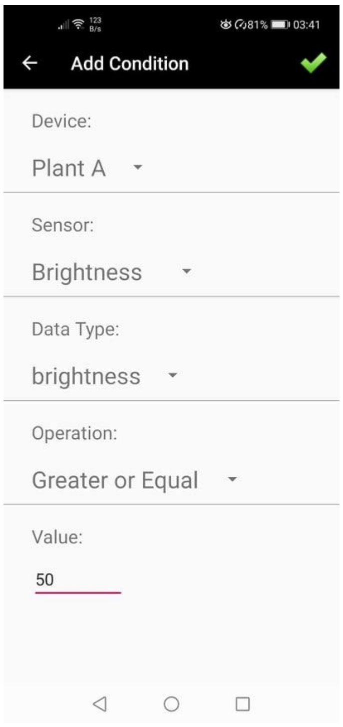
Trigger the following action(s)

Notification to email

Turn on Plant A's Front LED

Add an action

Figure 5.36 ConditionAction Activity



Add Condition

Device: Plant A

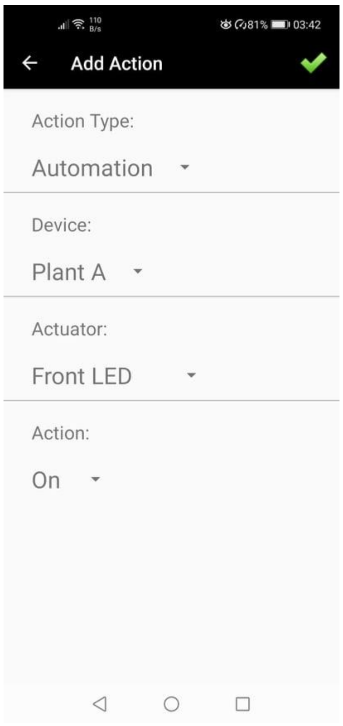
Sensor: Brightness

Data Type: brightness

Operation: Greater or Equal

Value: 50

Figure 5.37 Add Condition Activity



Add Action

Action Type: Automation

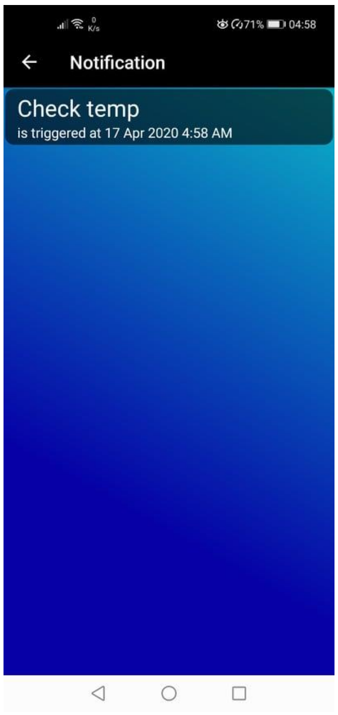
Device: Plant A

Actuator: Front LED

Action: On

Figure 5.38 Add Action Activity

When a trigger is triggered, the triggered timestamp will be stored as a historical record that can be viewed by users. Only the latest 30 history records can be viewed by users in the current project.



Notification

Check temp
is triggered at 17 Apr 2020 4:58 AM

Figure 5.39 Notification Activity

CHAPTER 6: CONCLUSION

6.1 Project Review and Discussion

IoT is blooming and changing the world by bringing thousand and one advantages to people. It changes the human living style from house, city, and eventually the world. With IoT technologies, a person can live in a smart house, in a safer city and a connected world. For instance, you can get a night of better sleep with a smart bed which will adjust the optimized temperature for the human body during the sleeping hour, you can avoid from traffic jams by the smart navigation system in your car, or even knowing the details of a product during shopping by wearing a smart glass. Besides changing the living style, other areas will be affected such as healthcare, industry, scientific and so on. The possibilities of IoT are only limited to human imagination and more advanced IoT technologies will be realized in the coming future.

The same knife cuts bread and finger. There are some existing problems in the IoT ecosystem. The problem statement of this project is discussing the challenges in customizing their IoT solution in a user-friendly manner. As mentioned by Kamat in 2017, people always demanding an IoT solution to perfectly solve their problems. There are normally 2 practices in customizing an IoT solution in the current ecosystem. First, find an IoT provider to customize the solution that raises the cost-effective issue. Another practice is to customize the solution by the IoT users. This practice is much more difficult for non-tech savvy user due to the huge learning gap in between. According to the research carried out by Metova in 2018, most of the IoT users are not even understanding IoT. This becomes a problem for the users to customize their solution from scratch.

There are some existing products in the market that try to solve the customisable issues. However, some applications such as IoTool, EzBlock Studio and RaspController are quite difficult to be used by novice users in IoT. As mentioned before in the problem statement, no matter how powerful is a product, when nobody knows how to use it, it is meaningless to have the product. Although these existing IoT products/platforms are powerful enough in terms of customisation, it is not friendly at all for those novice users to learn about IoT. For example, difficulty in device onboarding process, lack of monitoring features and weak user interface. In short, the main motivation of this project is to solve the customisation and user-friendliness issues in the IoT ecosystem in terms of monitoring, controlling, and connecting IoT devices.

Here comes with the proposed solution. The proposed solution of this project is to develop a customizable cloud-based IoT platform with user-friendly mobile interfaces in order to reduce the learning gap in customizing an IoT solution. At the end of the project, an Android application with Google Cloud platform supported is developed. By using the developed mobile application, users can easily connect to their devices and monitor the conditions in real-time. Moreover, users are able to customise their own IoT solution without much technical knowledge required by using the trigger feature in the mobile application. In a nutshell, both project objectives are achieved by providing a platform with a user-friendly mobile interface to customise their IoT solution easily.

6.2 Novelties and Contributions

The simple plug and play feature supported by the mobile interface is one of the main novelty in this project. With the immediate reflection in the user interface when users plug in or plug out a sensor/actuator from the devices, it coaches the users in understanding IoT concept and customising their own preferable solution. Last but not least, the advance controlling and monitoring features in the mobile application will further enhance the product’s usability and flexibility. Trigger feature is one of the advance controlling and monitoring features developed in this project. It allows users to have their own IoT solution easily by setting automation and notification rules. The advanced monitoring features such as the analytic graph view is able to assist users to derive insight into sensor data. For instance, users can roughly know how to set the condition parameters from past week average data. The automation rule set up in the trigger feature enhances the system’s flexibility when customising a solution as multiple conditions and actions can be set in a trigger. Other than that, data guarantee feature is also one of the unique points of this system. When users lost a sensor or a sensor is damaged, the sensor data collected before still able to continue with a new added sensor.

6.3 Future Work

Due to project time constraint, there are still a lot of future works can be done to improve the system. Although the current cloud architecture is able to support the project, problems may occur in the future as the IoT data will keep increasing. Firebase Realtime Database may become not efficient enough to handle a large amount of data. In the future, Google Cloud BigQuery may be used as a data warehouse and fast querying for IoT data. Other cloud tools such as Google Data Studio and Google ML kit can be used in the future in order to produce more value from collected sensor data. For example, suggest the best parameter for planting a potato by using Artificial Intelligent technology.

Other than that, the trigger feature can be improved as well by adding more parameters for users to customise their solution. For instance, condition checking interval, condition types, action types, action triggered period, types of notification and other advance automation rules. Currently, the system only support a predefined condition checking interval which defined in Cloud Scheduler. In the future, Cloud Scheduler API can be used to allow the user to define their own interval according to their preference. Besides, although the current system provides a notification feature to the user's email, more notification features should be provided in future work. Last but not least, there are some delay issues faced in the current video streaming module and efforts should be taken to improve streaming module performance and experience in future work.

BIBLIOGRAPHY

Android Developers. (n.d.). Meet Android Studio | Android Developers. [online] Available at: https://developer.android.com/studio/intro [Accessed 15 Aug. 2019].

Bedi, G., Venayagamoorthy, G. and Singh, R. (2016). Internet of Things (IoT) sensors for smart home electric energy usage management. 2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS). [online] Available at: https://ieeexplore.ieee.org/abstract/document/7946568/references#references [Accessed 10 Aug. 2019].

Corporation, H. (2019). *ON!Track Active Tracking*. [online] Hilti.group. Available at: https://www.hilti.group/content/hilti/CP/XX/en/services/tool-services/on-track/on-track-active-tracking.html#nav/close [Accessed 16 Nov. 2019].

Eliftech (2019). 7 Challenges of IoT Software Development. [online] IoT For All. Available at: https://www.iotforall.com/iot-software-development-challenges/ [Accessed 11 Aug. 2019].

Ettore Gallina. (2019). RaspController. Version 3.4.1. [Mobile app]. Google Play Store.

EZblock. (n.d.). What you can do with EzBlock Pi ?. [online] Available at: https://ezblock.raspad.com/ [Accessed 13 Aug. 2019]

Foote, K. (2016). A Brief History of the Internet of Things - DATAVERSITY. [online] DATAVERSITY. Available at: https://www.dataversity.net/brief-history-internet-things/ [Accessed 9 Aug. 2019].

GallinaEttore.com. (2019). RaspController - GallinaEttore.com. [online] Available at: https://www.gallinaettore.com/android_apps/raspcontroller/ [Accessed 15 Aug. 2019].

Google Cloud. (n.d.). What Is Cloud Pub/Sub? | Cloud Pub/Sub Documentation | Google Cloud. [online] Available at: https://cloud.google.com/pubsub/docs/overview [Accessed 15 Aug. 2019].

Hackbarth, K. (2016). The three challenges of IoT solution development - Bosch ConnectedWorld Blog. [online] Bosch ConnectedWorld Blog. Available at: https://blog.bosch-si.com/internetofthings/the-three-challenges-of-iot-solution-development/ [Accessed 16 Nov. 2019].

Hilti Corporation. (n.d.). Internet of Things. [online] Hilti.group. Available at: https://www.hilti.group/content/hilti/CP/XX/en/services/tool-services/internet-of-things.html#nav/close [Accessed 10 Aug. 2019].

i-SCOOP. (n.d.). The Internet of Things in manufacturing: benefits, use cases and trends. [online] Available at: https://www.i-scoop.eu/internet-of-things-guide/internet-of-things-in-manufacturing/ [Accessed 10 Aug. 2019].

Iotinabox.com. (n.d.). IoT in a Box – Turnkey IoT Solutions. [online] Available at: https://www.iotinabox.com/ [Accessed 14 Aug. 2019].

IoTool. (n.d.). Home - IoTool. [online] Available at: https://iotool.io/#dashboard [Accessed 14 Aug. 2019].

Kamat, H. (2017). What is the Custom IoT Solutions?. [online] Quora.com. Available at: https://www.quora.com/What-is-the-Custom-IoT-Solutions [Accessed 11 Aug. 2019].

Kickstarter. (n.d.). Ezblock Pi - The Ultimate Companion Board for RPi Projects!. [online] Available at: https://www.kickstarter.com/projects/35410622/ezblock-pi-the-ultimate-companion-board-for-rpi-projects?fbclid=IwAR2pd1h6TaT-pS9sAiIOcOWXmEPii6chbFZJTtOSTE9hbVmd2gbPr4ghFGw [Accessed 13 Aug. 2019].

Krupitzer, C. (n.d.). Building a Custom IoT implementation Isn’t as Difficult as it Seems – ThingLogix, Inc. [online] Thinglogix.com. Available at: https://www.thinglogix.com/building-a-custom-iot-implementation-isnt-as-difficult-as-it-seems/ [Accessed 11 Aug. 2019].

Lajartre, M. (2019). Infographic: The Internet of Things (IoT) is a booming business. [online] Cpacanada.ca. Available at: https://www.cpacanada.ca/en/news/world/2019-02-13-internet-of-things-infographic [Accessed 16 Nov. 2019].

Lego.com. (n.d.). *About | Mindstorms | Official LEGO® Shop MY*. [online] Available at: https://www.lego.com/en-my/themes/mindstorms/about [Accessed 21 Dec. 2019].

Manhattan Street Capital. (n.d.). Skydrop™. [online] Available at: https://www.manhattanstreetcapital.com/skydrop%E2%84%A2 [Accessed 14 Aug. 2019].

McKendrick, J. (2018). Is The Internet of Things Growing Too Fast? - RTInsights. [online] RTInsights. Available at: https://www.rtinsights.com/is-the-internet-of-things-growing-too-fast/ [Accessed 9 Aug. 2019].

Metova. (2018). Infographic - The Internet of Things - Metova. [online] Available at: https://metova.com/infographic-the-internet-of-things/ [Accessed 16 Nov. 2019].

Mohanraj, I., Ashokumar, K. and Naren, J. (2016). Field Monitoring and Automation Using IOT in Agriculture Domain. Procedia Computer Science, 93, pp.931-939.

Morgan, J. (2014). A Simple Explanation Of 'The Internet Of Things'. [online] Forbes.com. Available at: https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#266c753d1d09 [Accessed 9 Aug. 2019].

myDevices. (2019). IoT in a Box. Version 2.5.0. [Mobile app]. Google Play Store.

My.mouser.com. (n.d.). *SensorTag Kits - TI | Mouser Malaysia*. [online] Available at: https://my.mouser.com/new/texas-instruments/ti-sensor-tag-kits/ [Accessed 22 Dec. 2019].

Pande, S. (2017). What is the Custom IoT Solutions?. [online] Quora.com. Available at: https://www.quora.com/What-is-the-Custom-IoT-Solutions [Accessed 11 Aug. 2019].

Petrov, C. (2019). 40 Internet Of Things Statistics from 2019 to Justify The Rise of IoT. [online] Tech Jury. Available at: https://techjury.net/stats-about/internet-of-things-statistics/ [Accessed 11 Aug. 2019].

Ranger, S. (2018). What is the IoT? Everything you need to know about the Internet of Things right now | ZDNet. [online] ZDNet. Available at: https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/ [Accessed 9 Aug. 2019].

Rouse, M. (n.d.). What is internet of things (IoT)? - Definition from WhatIs.com. [online] IoT Agenda. Available at:

https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT [Accessed 9 Aug. 2019].

Skydrop.com. (n.d.). Home Owners | Skydrop - Smart Sprinkler Controller. [online] Available at: https://www.skydrop.com/home-owners/ [Accessed 14 Aug. 2019].

StoneFly. (n.d.). The Role of Cloud Computing in the Internet of Things. [online] Available at: https://stonefly.com/blog/role-cloud-computing-internet-things [Accessed 12 Aug. 2019].

SunFounder. (2019). Ezblock Studio. Version 0.1.0. [Mobile app]. Google Play Store.

Texas Instruments. (n.d.). *SimpleLink™ multi-standard CC2650 SensorTag™ kit reference design TIDC-CC2650STK-SENSORTAG*. [online] Available at: http://www.ti.com/tool/TIDC-CC2650STK-SENSORTAG# [Accessed 22 Dec. 2019].

Tzounis, A., Katsoulas, N., Bartzanas, T. and Kittas, C. (2017). Internet of Things in agriculture, recent advances and future challenges. Biosystems Engineering, 164, pp.31-48.

Xu, L., Xu, E. and Li, L. (2018). Industry 4.0: state of the art and future trends. International Journal of Production Research, 56(8), pp.2941-2962.

Yang, C., Lan, S., Shen, W., Huang, G., Wang, X. and Lin, T. (2017). Towards product customisation and personalization in IoT-enabled cloud manufacturing. Cluster Computing, 20(2), pp.1717-1730.

APPENDICES - WEEKLY LOG

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 1**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE Report on current progress of FYP 2 during semester break. During semester break, more researches are carried out to review other IoT Products including Lego Mindstorm EV3 and SimpleLink SensorTag. Discuss with supervisor regarding the implementation of analytic graph feature.
2. WORK TO BE DONE Start coding for analytic graph feature.
3. PROBLEMS ENCOUNTERED Faces difficulty in researching the IoT products as the products need to be purchased for detailed review purpose. It is difficult to know the details of an IoT product by researching online only.

4. SELF EVALUATION OF THE PROGRESS Should be more active during semester break.

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 2**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE Completion of analytic graph feature including the mobile application development and Google Cloud Function development. Discuss with supervisor regarding the feature and progress direction.
2. WORK TO BE DONE Fine tune the analytic graph feature. Start working in add device module development (Mobile Application Development).
3. PROBLEMS ENCOUNTERED Synchronisation issue is faced when dealing with Firebase Realtime Database and Google Cloud Function.

4. SELF EVALUATION OF THE PROGRESS

Good progress in completing the analytic graph feature before the scheduled milestone.

[illegible]

Supervisor’s signature

[illegible]

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 3**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE Partially completion of add device module – QR code scanning feature is added in add device module. Users are allowed to scan the QR code and retrieve the information from the QR code.
2. WORK TO BE DONE Start researching and working on existing protocol in order to allow users to add device by the combination of protocol and QR code scanner.
3. PROBLEMS ENCOUNTERED After completion of QR code scanning feature, the progress stucks due to the device onboarding procedure is still not clear enough. More brainstorming is needed to be carried out. How to ease the onboarding process as much as possible yet still maintain the security issue of connecting IoT device ? Besides, lacking of knowledge in existing protocol that allow users to add a device by scanning QR code has become one of the problems faced in this week.

4. SELF EVALUATION OF THE PROGRESS

More works and researches are needed to be carried out.

[illegible]

Supervisor’s signature

[illegible]

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 4**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE After some researches on existing protocol such as Bluetooth and Wifi, Bluetooth protocol is decided to be used in this project. The device onboarding procedure is decided after brainstorming with supervisor and project teammate. Start to code for Bluetooth protocol in mobile application.
2. WORK TO BE DONE Complete the add device module by next week.
3. PROBLEMS ENCOUNTERED It is difficult to decide which protocol can be used in this project as both project members do not have any experience in the protocols such as Bluetooth and Wifi. There are some concerns need to be considered such as project time constraint, protocol's suitability and the complexity of adopting the protocol.

4. SELF EVALUATION OF THE PROGRESS
Lacking of knowledge in existing protocols has delayed some project development progress. More researches should be carried out as earlier as possible.

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 5**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE Completed Bluetooth protocol development and able to combine the protocol with QR scanner developed. Users are allowed to add a device by scanning Nearby Bluetooth devices or scan the QR code of the device. Both methods are using same Bluetooth protocol.
2. WORK TO BE DONE Start to plan on how to allow users to set some simple logic with parameter on multiple devices in order to enhance system customisation ability.
3. PROBLEMS ENCOUNTERED The Bluetooth protocol is not stable enough. For example, nearby Bluetooth devices may not be found during first time scanning. Users may need to scan again the nearby Bluetooth devices. It caused some problems when combining QR scanner with Bluetooth protocol.

4. SELF EVALUATION OF THE PROGRESS Good progress in this week and able to complete the deadline set for developing add device module.

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 6**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE Small adjustment done in add device module. Able to figure out a few approaches to allow users to set simple if-then logic on multiple devices added.
2. WORK TO BE DONE After discussion with supervisor, an approach is decided to develop the trigger feature by using Cloud Scheduler. Start implementing the approach.
3. PROBLEMS ENCOUNTERED I have spent a lot of time in brainstorming in order to figure out a best approach to develop the trigger features including front end and back end logic design.

4. SELF EVALUATION OF THE PROGRESS Spent too much time in brainstorming the approach.

[illegible]

Supervisor’s signature

[illegible]

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 7**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE

Start to code the trigger feature's backend logic in Cloud Function.

2. WORK TO BE DONE

Complete all the back end development of the trigger feature and start working on front end development by next week.

3. PROBLEMS ENCOUNTERED

After the approach is decided, the development complexity is still a challenge to realise the approach.

There are a lot of technical issues found when developing the back end logic due the complexity of coding logic.

4. SELF EVALUATION OF THE PROGRESS

The trigger feature backend development progress must be finished by next week as there are still a lot of works need to be done.

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 8**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE Able to complete the trigger back end development by using Cloud Function and Cloud Scheduler. A Cloud Function is created to monitor multiple conditions in a trigger and perform corresponding actions.
2. WORK TO BE DONE Start working on front end user interface of trigger feature.
3. PROBLEMS ENCOUNTERED A lot of time spent for coding and debugging the Cloud Function.

4. SELF EVALUATION OF THE PROGRESS Need to speed up the progress

[illegible] [illegible]

Supervisor’s signature Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 9**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE

User interface of trigger features in mobile application is developed.

The overall trigger feature is completed.

Users are allow to add, edit, retrieve and update the trigger from mobile application.

2. WORK TO BE DONE

After discussion with supervisor, simple notification feature is needed when the trigger is triggered.

3. PROBLEMS ENCOUNTERED

Combining front end interface (mobile application) and back end (Google Cloud Function) spent almost 1 weeks to be finished.

Sometime, the development of interface and back end logic need to be adjusted even though draft designs are done before coding.

4. SELF EVALUATION OF THE PROGRESS

Good progress in this week. The project progress is on track.

[illegible]

Supervisor’s signature

[illegible]

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 10**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE Completed notification module. Users are allowed to send notification to their email if a trigger is triggered. Besides, trigger history will be available for users to view last 30 records. Completed overall trigger module with user interfaces, Google Cloud Function and Cloud Scheduler.
2. WORK TO BE DONE Testing and debugging overall system.
3. PROBLEMS ENCOUNTERED Project due date is around the corner, need to speed up development progress. No sufficient time to optimise the trigger module.

4. SELF EVALUATION OF THE PROGRESS

Good progress in this week but need to start testing as soon as possible.

[illegible] [illegible]

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 11**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE Found bugs in Bluetooth add device module and able to fix the bug. The overall system user interface is improved as well in this week.
2. WORK TO BE DONE Start preparing for report and product demonstration video clip.
3. PROBLEMS ENCOUNTERED The Bluetooth bug occurred after updating the smartphone system and it spent a lot of time in debugging and fixing process.

4. SELF EVALUATION OF THE PROGRESS Need to continue the efforts as the project deadline is coming soon.

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 12**
Student Name & ID: Ang Cheng Kee 1603121	
Supervisor: Dr Cheng Wai Khuen	
Project Title: Customisable Cloud-Based IoT Monitoring System with Smartphone Interface	

1. WORK DONE

Developed a small feature in Cloud Function to ensure database storage efficiency.

Completed the draft report and submit to supervisor for checking.

Video clip production is still in progress.

2. WORK TO BE DONE

Complete the video clip and prepare for presentation.

Fine tune the report if any changes required.

Finalised the report.

3. PROBLEMS ENCOUNTERED

Combining front end interface (mobile application) and back end (Google Cloud Function) spent almost 1 weeks to be finished.

Sometime, the development of interface and back end logic need to be adjusted even though draft designs are done before coding.

4. SELF EVALUATION OF THE PROGRESS Good progress in this week. The project progress is on track.

[illegible] [illegible]

Supervisor’s signature Student’s signature

POSTER

EAZYIOT

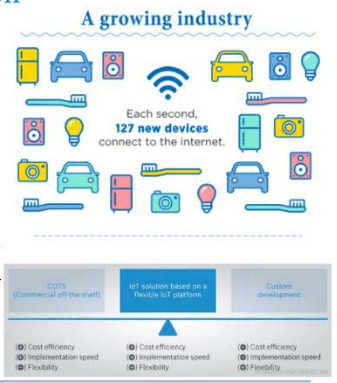
A Customisable Cloud-Based IoT Monitoring System with Smartphone Interface

By Ang Cheng Kee

1 Introduction

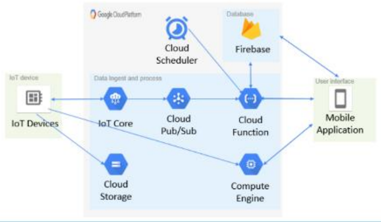
Internet of Things (IoT), a growing Industry that changes our world. With IoT, things around us are sharing data and form a connected ecosystem. People tend to make use of IoT to modernize living environments such as smart home, smart city, and others.

However, problems raised with IoT as well. Customisation, user-friendliness, learning gap and data monitoring have become one of the biggest concerns for IoT users. The existing IoT market is lacking of a flexible IoT platform to support users in learning, monitoring and customising their solution from a user-friendly interface.



2 Proposed Method

The proposed method is to develop a customisable cloud-based IoT monitoring system with user -friendly interfaces provided to support IoT devices connection, controlling and monitoring. With the interfaces provided, IoT users are allowed to having their own preferable solutions easily by reducing the huge learning gap in between. Besides, the proposed method will act as a coaching tool for novice IoT users in learning and customising their IoT solutions.



3 Result and Discussion

Single Interface, Multiple Things

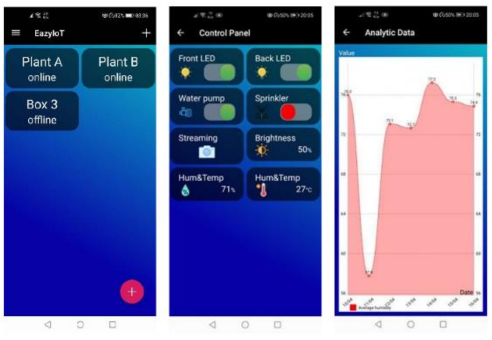
- Quick and easy setup for multiple devices
- Controlling and monitoring devices through a single interface

Real-Time Monitoring

- Visualise sensor data with real-time update
- Trigger action for condition set
- Send notification to users

Additional Features

- Visualise and customise analytic graph
- Live video streaming with camera module



4 Conclusion

Quick Steps to build your IoT Solution:

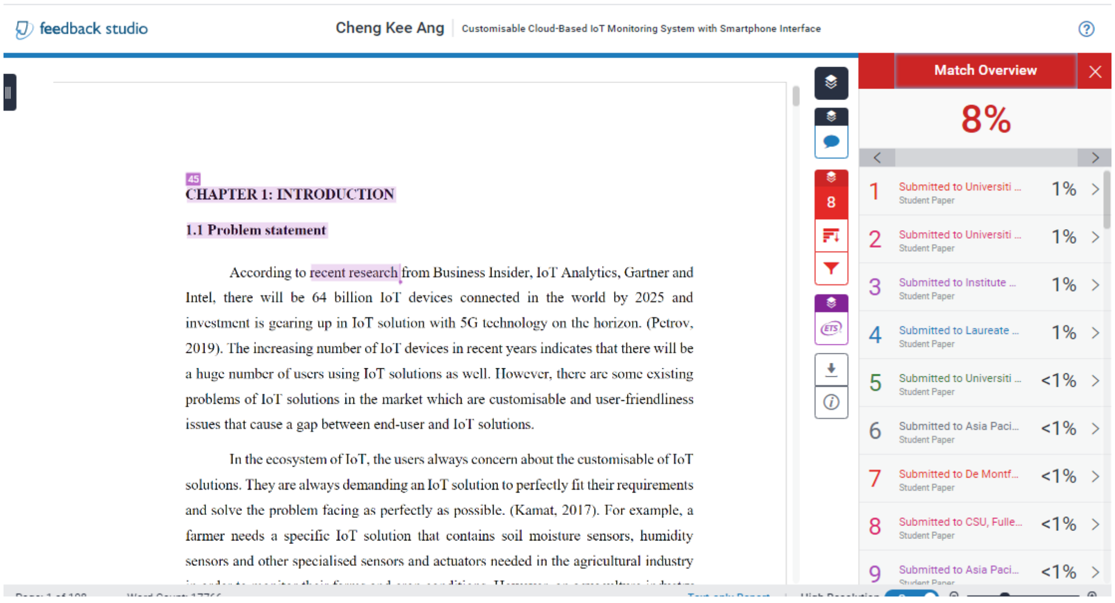
1. Purchase IoT devices.
2. Download mobile application.
3. Connect to the devices and start to monitor and build your own solutions.

In short, EazyIoT allows users to learn, customise, and build their solutions with a minimum effort.

WITH EAZYIOT, EVERYTHING BECOMES EAZY.

,

PLAGIARISM CHECK RESULT



feedback studio   Cheng Kee Ang   Customisable Cloud-Based IoT Monitoring System with Smartphone Interface

45

CHAPTER 1: INTRODUCTION

1.1 Problem statement

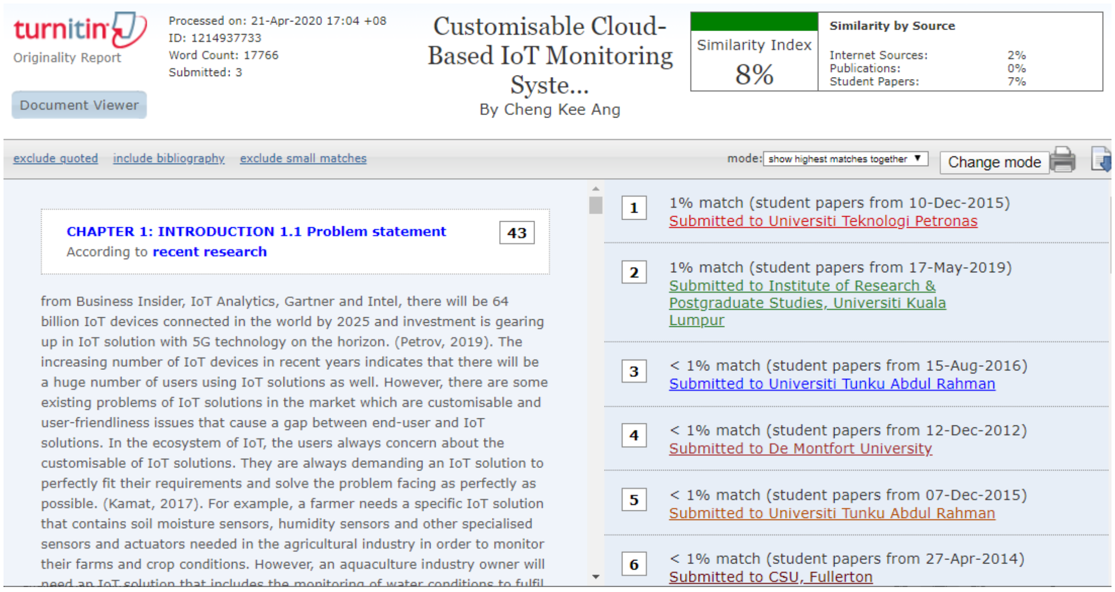
According to recent research from Business Insider, IoT Analytics, Gartner and Intel, there will be 64 billion IoT devices connected in the world by 2025 and investment is gearing up in IoT solution with 5G technology on the horizon. (Petrov, 2019). The increasing number of IoT devices in recent years indicates that there will be a huge number of users using IoT solutions as well. However, there are some existing problems of IoT solutions in the market which are customisable and user-friendliness issues that cause a gap between end-user and IoT solutions.

In the ecosystem of IoT, the users always concern about the customisable of IoT solutions. They are always demanding an IoT solution to perfectly fit their requirements and solve the problem facing as perfectly as possible. (Kamat, 2017). For example, a farmer needs a specific IoT solution that contains soil moisture sensors, humidity sensors and other specialised sensors and actuators needed in the agricultural industry

Match Overview

8%

1	Submitted to Universiti ... Student Paper	1%
2	Submitted to Universiti ... Student Paper	1%
3	Submitted to Institute ... Student Paper	1%
4	Submitted to Laureate ... Student Paper	1%
5	Submitted to Universiti ... Student Paper	<1%
6	Submitted to Asia Paci... Student Paper	<1%
7	Submitted to De Montf... Student Paper	<1%
8	Submitted to CSU, Fulle... Student Paper	<1%
9	Submitted to Asia Paci... Student Paper	<1%



turnitin Originality Report

Processed on: 21-Apr-2020 17:04 +08
ID: 1214937733
Word Count: 17766
Submitted: 3

Customisable Cloud-Based IoT Monitoring Syste...
By Cheng Kee Ang

Similarity Index 8%

Similarity by Source

Internet Sources:	2%
Publications:	0%
Student Papers:	7%

Document Viewer

exclude quoted   include bibliography   exclude small matches   mode: show highest matches together   Change mode

CHAPTER 1: INTRODUCTION 1.1 Problem statement According to recent research 43

from Business Insider, IoT Analytics, Gartner and Intel, there will be 64 billion IoT devices connected in the world by 2025 and investment is gearing up in IoT solution with 5G technology on the horizon. (Petrov, 2019). The increasing number of IoT devices in recent years indicates that there will be a huge number of users using IoT solutions as well. However, there are some existing problems of IoT solutions in the market which are customisable and user-friendliness issues that cause a gap between end-user and IoT solutions. In the ecosystem of IoT, the users always concern about the customisable of IoT solutions. They are always demanding an IoT solution to perfectly fit their requirements and solve the problem facing as perfectly as possible. (Kamat, 2017). For example, a farmer needs a specific IoT solution that contains soil moisture sensors, humidity sensors and other specialised sensors and actuators needed in the agricultural industry in order to monitor their farms and crop conditions. However, an aquaculture industry owner will

1 1% match (student papers from 10-Dec-2015)
Submitted to Universiti Teknologi Petronas

2 1% match (student papers from 17-May-2019)
Submitted to Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur

3 < 1% match (student papers from 15-Aug-2016)
Submitted to Universiti Tunku Abdul Rahman

4 < 1% match (student papers from 12-Dec-2012)
Submitted to De Montfort University

5 < 1% match (student papers from 07-Dec-2015)
Submitted to Universiti Tunku Abdul Rahman

6 < 1% match (student papers from 27-Apr-2014)
Submitted to CSU, Fullerton

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Ang Cheng Kee
ID Number(s)	16ACB03121
Programme / Course	Bachelor of Computer Science (Hons)
Title of Final Year Project	Customisable Cloud-Based IoT Moniroting System with Smartphone Interface

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 8 **%** **Similarity by source** Internet Sources: 2 % Publications: 0 % Student Papers: 7 %	no issue, ok.
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

[illegible]

Signature of Supervisor	Signature of Co-Supervisor
Name: Dr. Cheng Wai Khuen	Name:
Date: 23 April 2020	Date:

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	16ACB03121
Student Name	Ang Cheng Kee
Supervisor Name	Dr. Cheng Wai Khuen

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
/	Front Cover
/	Signed Report Status Declaration Form
/	Title Page
/	Signed form of the Declaration of Originality
/	Acknowledgement
/	Abstract
/	Table of Contents
/	List of Figures (if applicable)
/	List of Tables (if applicable)
/	List of Symbols (if applicable)
/	List of Abbreviations (if applicable)
/	Chapters / Content
/	Bibliography (or References)
/	All references in bibliography are cited in the thesis, especially in the chapter of literature review
/	Appendices (if applicable)
/	Poster
/	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. [illegible] ______________________ (Signature of Student) Date: 21 April 2020	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. [illegible] ______________________ (Signature of Supervisor) Date: 23 April 2020