THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY,VELLORE.

IOT BASED SAFETY GADGET FOR CHILD SAFETY MONITORING &NOTIFICATION

DOMAIN:INTERNET OF THINGS

TEAM MEMBERS:

- SUBALAKSHMI G - 513119106314
- VELMURUGAN V- 513119106317
- RANJITHKUMAR R - 513119106311
- TAMIZH D - 513119106316

1.INTRODUCTION

1.1 OVERVIEW

This paper presents the design and implementation of a portable IOT-based safety and health monitoring system for children through a sensor embedded health monitoring device for safety and emergency services. It is known that the technological advancements are increasing at a faster pace. But the utilization of technologies in various sectors is very low. We know that people of different age group faces different difficulties. But the security for children‟s is very low. There is lot of cases registered regarding child safety.

1.2 PURPOSE

Nowadays, the schools and the parents are very much worried about their school children‟s for school transport and other places. So, the safety and monitoring the school children is very much difficult. In this project we are introducing the IOT based embedded system is used in this project. So we propose a system to continuously monitor the parameters of the child and also their location for safety purpose. The system provides smart child tracking and monitoring system.

2 LITERATURE SURVEY

PAPER 1: Smart Waste Management System Using LoRa and Tensorflow Deep Learning Model

Publication year: August 12, 2020

Author name: TEOH JI SHENG 1 , MOHAMMAD SHAHIDUL ISLAM 1 , (Graduate Student Member, IEEE), NORBAHIAH MISRAN 1 , (Senior Member, IEEE), MOHD HAFIZ BAHARUDDIN 1 , (Member, IEEE), HASLINA ARSHAD 2 , MD. RASHEDUL ISLAM 1 , MUHAMMAD E. H. CHOWDHURY 3 , (Member, IEEE), HATEM RMILI 4 , (Senior Member, IEEE), AND MOHAMMAD TARIQUL ISLAM 1 , (Senior Member, IEEE).

Summary: Traditional waste management system operates based on daily nd allow for better waste management. The aim of this research is to develop a smart waste management system using LoRa communication protocol and TensorFlow based deep learning model. LoRa sends the sensor data and Tensorflow performs real time object detection and classification. The bin consists of several compartments to segregate the waste including metal, plastic, paper, and general waste compartment which are controlled by the servo motors. Object detection and waste classification is done in TensorFlow framework with pre-trained object detection model. This object detection model is trained with images of waste to generate a frozen inference graph used for object detection which is done through a camera connected to the Raspberry Pi 3 Model B+ as the main processing unit. Ultrasonic sensor is embedded into each waste compartment to monitor the filling

level of the waste. GPS module is integrated to monitor the location and real time of the bin. LoRa communication protocol is used to transmit data about the location, real time and filling level of the bin. RFID module is embedded for the purpose of waste management personnel identification.

Methodology used: This work was supported in part by the Ministry of Malaysia under Grant LRGS MRUN/F2/01/2019/1/2, and in part by the Deanship of Scientific Research (DSR), King Abdulaziz Universityy, Jeddah, Saudi Arabia, under Grant RG-39-135-40.

Conclusion: This article presented a smart waste management system by implementing sensors to monitor the status of the bin, LoRa communication protocol for low power and long-range data transmission, and TensorFlow?based object detection to perform waste identification and classification. The pre-trained object detection model, SSDMobilnetV2 is able to perform well in Raspberry Pi 3 Model B+ due to its lightweight nature. The model was able to detect and classify waste according to classes such as metal, plastic, and paper. However, the accuracy of the model can be improved by increasing the number of training data—in this case, the number of waste images—and by increasing the training time. The segregation of waste is interfaced and coordinated well between the object detection performed by Raspberry Pi and the servo motor controlling the lid of the individual waste compartment. An RFID module controls the locking mechanism of the bin. Ultrasonic sensors monitor the filling level, while the GPS module monitors the location and real-time of the bin. LoRa operating at a frequency band of 915MHz transmits data regarding the status of the bin regarding filling level, location, and real-time from the bin to the LoRa gateway. The data received at the gateway is decoded by a terminal program, RealTerm. This automated segregation and monitoring system implementation in the bin aims to reduce the operating cost and improve the waste management system. At the same time, we are eager to develop the city into asmart city. In the future, the waste detection model is to be improved by increasing the number of waste images in the dataset to increase the flexibility of the system in identifying waste. Moreover, an automated routing system can be developed to identify and pinpoint the shortest path to the bin for the purpose of maintenance. With this in mind, the existing waste management system can be improved and bring society towards a greener and healthier life.

PAPER 2: Smart City Platform Environment for Waste Management

Publication year: 05 | May 2019

Author name: G. Paulin Nancy1, R. Resmi2 Journel name: International Research Journal of Engineering and Technology (IRJET)

Summary: Coimbatore city is one of the smart cities. There are many projects going on for the development of Coimbatore as a smart city. Waste management has become a challenge before society as it is being continuously neglected in the field of environment which is getting harmful for the health of living organism's as well as the environment. Effective waste management strategies are required that involves a synchronized system of controlling the production and disposal of wastes. Most of the waste management techniques like landfills, incineration, sanitary landfills provide a variety of environmental benefits but have negative impacts too like emission of large amount of green house gas. This paper reveals the risk and issues occurred during all stages of waste management and find the smart solution for those major issues thereby developing the platform of smart city for waste management.

Methodology used: Identifying the key risk factors of waste management process by reviewing the literature and through the additions that could be made by the participants i.e. workers .Questions are prepared based on the identified risk factors such as storage system, lack of proper segregation, area coverage, capacity issue, climatic change, etc. Major factors and issues are identified with the help of questionnaire survey. Providing practical suggestions and recommendations pointing toward upgrading waste management process and improve the performance of workers thereby create platform of smart city.

Conclusion: From the survey report, several risk factors such as storage system, lack of proper segregation, area coverage, capacity issue, climatic change were identified. Based on these risk factors, smart solutions recommended were automated sorting system, automated solid waste management, smart planning by using web camera & load sensor, Smart garbage.

PAPER 3: IoT-based smart waste level monitoring system for smart cities

Publication year: January 2021

Author name: A.A.I. Shah1 , S.S.M. Fauzi2 , R.A.J.M. Gining3 , T.R.Razak4 , M.N.F.Jamaluddin5 , R. Maskat6 Journel name: Indonesian Journal of Electrical Engineering and Computer Science

Summary: Smart cities are covering the population that are seeking the best lifestyle and fulfilling their needs. Through smart cities, necessary modern facilities using ICT emerging technologies such as the internet of things (IoT) had been installed to ensure the sustainability of the city. In the perspective of waste management, several different IoT-based solutions also had been proposed as an alternative to monitor and to ensure the health of communities. This paper reviews existing IoT-based solutions in smart cites' waste level management system to bring together the state-of-the-art. We performed reviews on 16 research articles from the past 5 years in the literature to provide a comprehensive review

of different works on IoT-based solutions related to the smart waste level monitoring system, possible solutions and technologies used. The results obtained shows that existing solutions were similar in the platform used to integrate with the IoT technologies but had some differences in term of the used of sensors and communication technologies. The study also shows that many of the prior studies used Arduino Uno. Results from this study will assist the researcher, focusing on expanding further the used of different technologies or improved the existing system

Methdology used: This study employs a necessary systematic mapping study (SMS) steps [35]. An SMS is intended to encompass an exhaustive search. It aims to provide a thorough and repeatable analysis of all relevant literature. The five main steps in the method are: definition of research questions, searching for relevant papers, screening papers, keywording of abstracts, and data extraction and mapping

Conclusion: This study contributes to research on smart waste level monitoring system by synthesising the literature on the current state-of-the-art. This study is crucial as it provides a clear overview of the state-ofthe-art of the development and implementation of the smart waste level monitoring system. An in-depth review suggests that the existing solutions were similar in the platform used to integrate with the IoT technologies but have some differences in term of the used of sensors and communication technologies. The study also shows that many of the prior studies used Arduino Uno. In future research, we intend to identify the requirement of the proposed

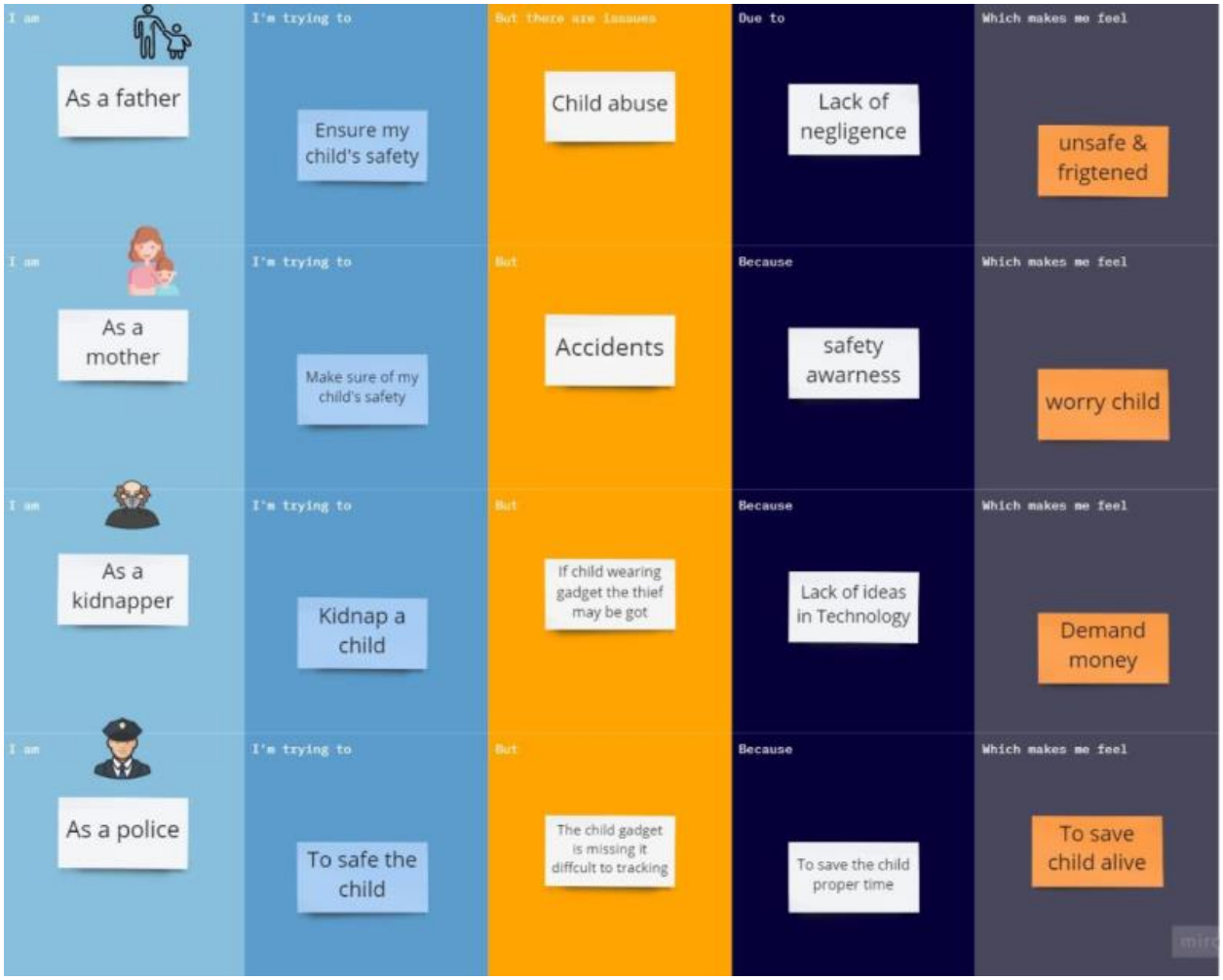
2.1 EXISTING PROBLEM

This system requires network connectivity, satellite communication, and high-speed data connection when we use web camera and GPS to lively monitor. **It is difficult to monitor when there occurs any hindrance to satellite communication or any network issue**.

2.2 REFERENCES

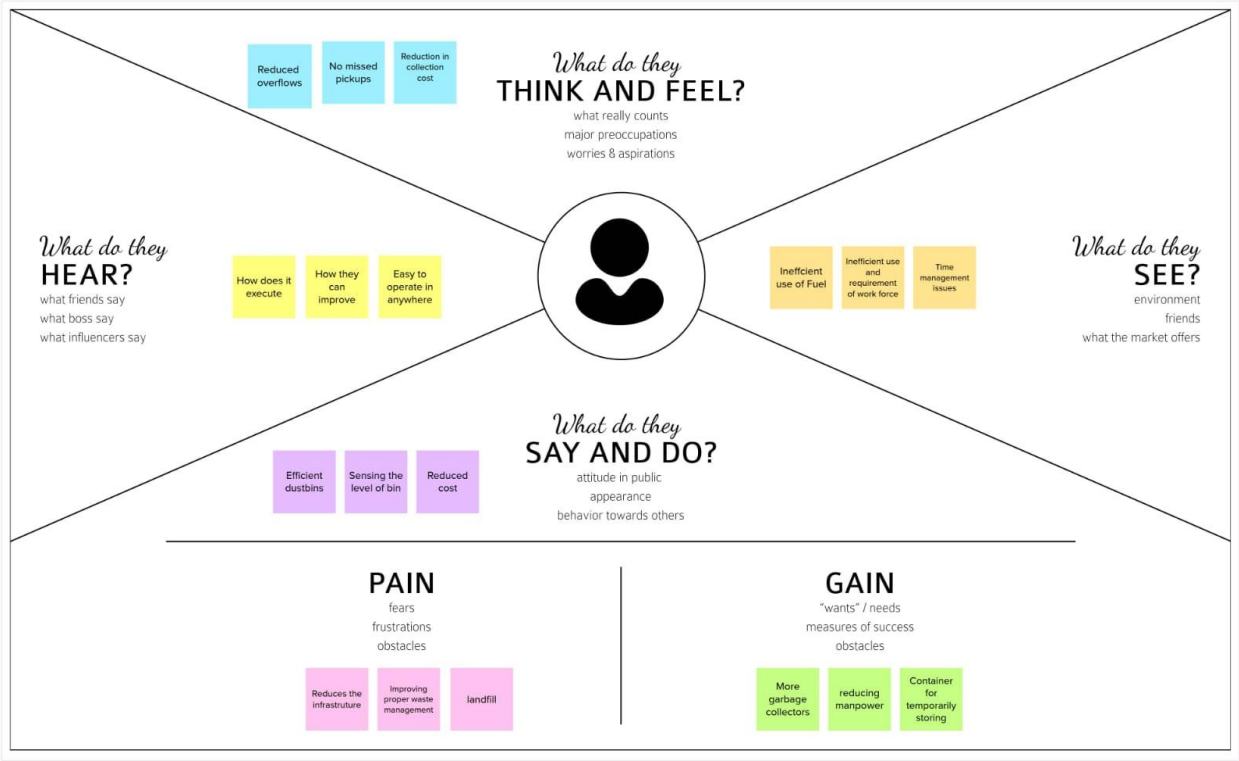
Child safety wearable device | IEEE Conference Publication | IEEE Xplore
RFID-based system for school children transportation safety enhancement | IEEE Conference Publication | IEEE Xplore
Design and development of an IOT based wearable device for the safety and security of women and girl children | IEEE Conference Publication | IEEE Xplore

2.3 PROBLEM STATEMENT DEFINITION



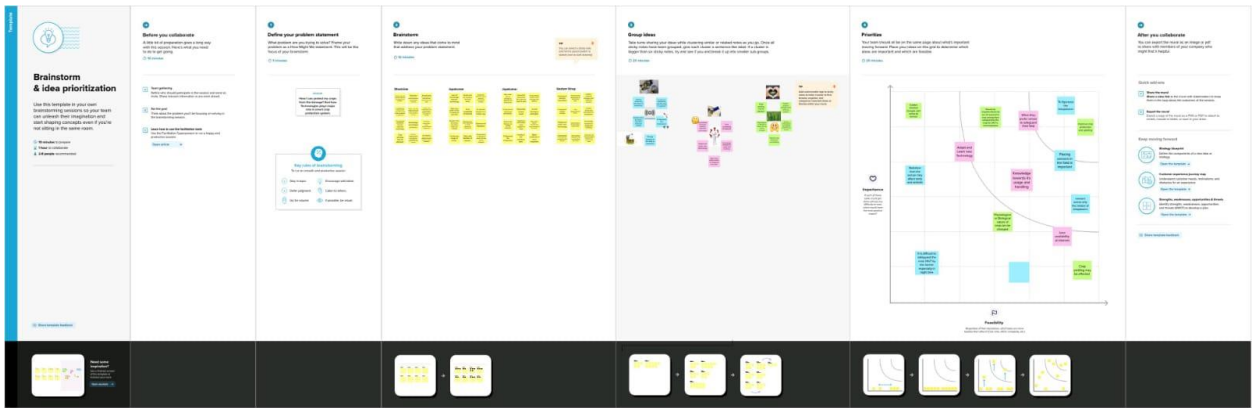
3.IDEATION &PROPOSED SOLUTION

3.1 Empathy map Canvas



[IBM-Project-14883-1659591535/Empathy map.pdf at main · IBM-EPBL/IBM-Project-14883-1659591535 · GitHub]

3.2 Ideation & BrainStorming



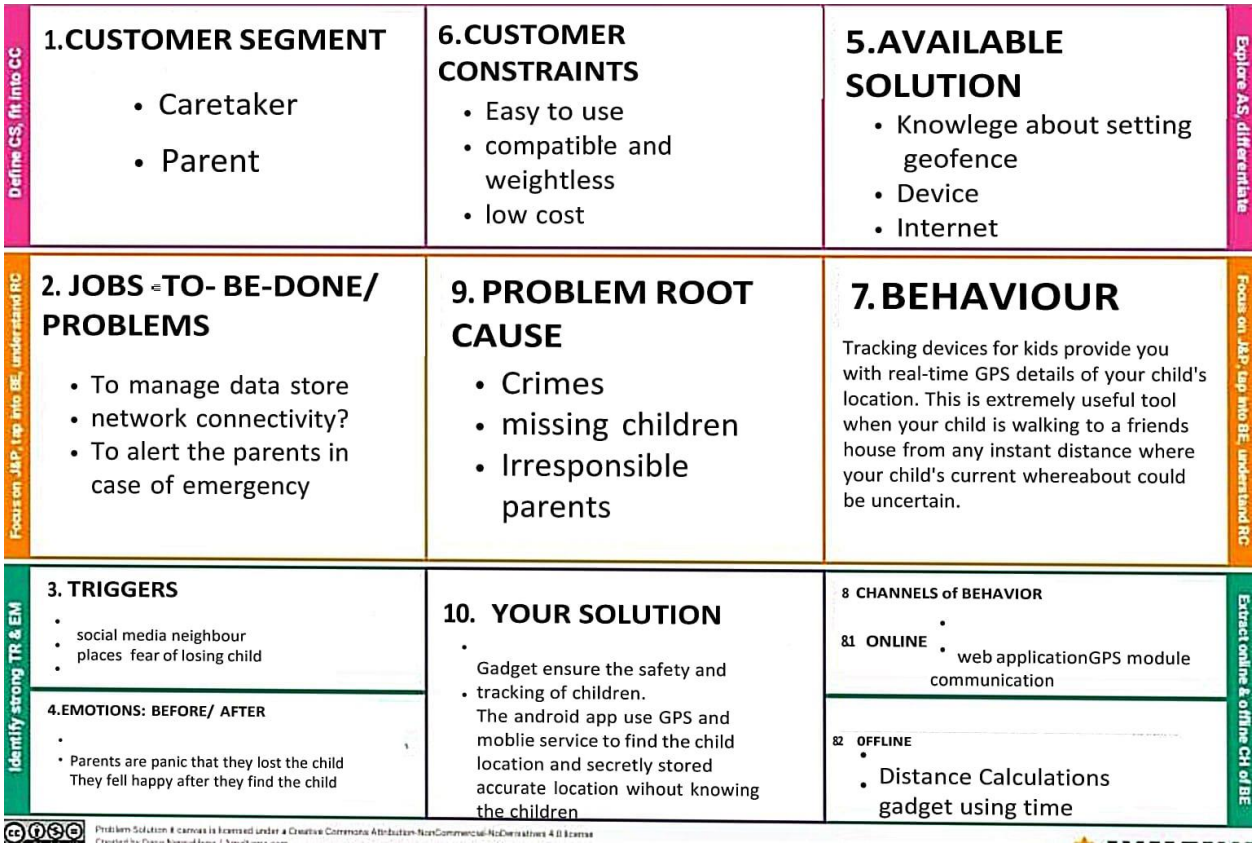
IBM-Project-14883-1659591535/IBM BRAINSTORMING.pdf at main · IBM-EPBL/IBM-Project-14883-1659591535 · GitHub

3.3 Proposed Solution

S.NO	PARAMETER	DESCRIPTION
1`	Problem Statement (Problem Solved)	Safety of children is very critical since children cannot protect themselves. A momentary lack in parental supervision should be combated with an appropriate IT solution in context
2	Idea/Solution description	In our system, we automatically monitor the child in real time using Internet of Things, with the help of GPS, GSM. This system requires network connectivity, satellite communication, and high-speed data connection when we use web camera and GPS to lively monito
3	Novelty/Uniqueness	The novelty of the work is that the system automatically alerts the parent/caretaker by sending SMS, when immediate attention is required for the child during

		emergency. The parameters such as touch, temperature &heartbeat of the child are used for parametric analysis and results are plotted for the same
4	Social Impact/Customer Satisfaction	Safer society for the children. Receive immediate information about child and feel secure.
5	Business Model(Revenue Model)	Prevention against immediately and safely if in dang
6	Scalability of the SDolution	To monitor health conditions of a child (heart rate, body temperature, body posture).

3.4 PROBLEM SOLUTION FIT



4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

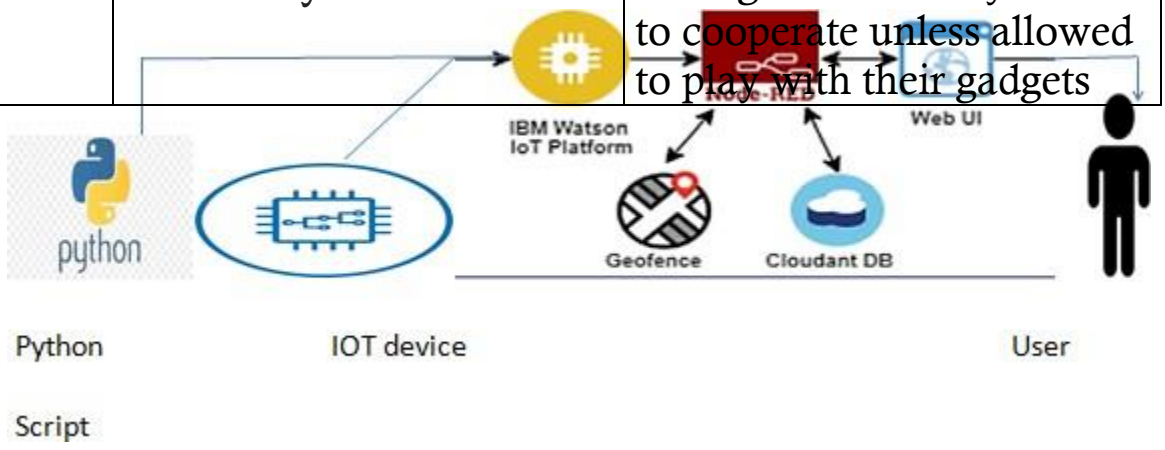
FR.NO	Functional Requirement	Sub Requirement(Story/Sub-Task)
FR-1	Communicate and exchange information to provide server for user	1. To monitor the children's location continuously in schools or parks. 2. Alert the parent if the child crossesthe geofence through

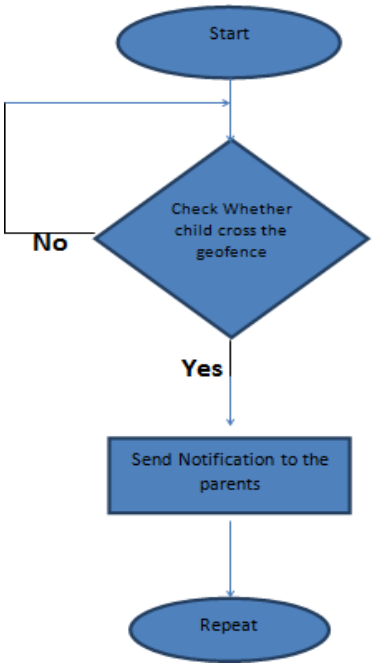
		SMS.
FR-2	Continuous monitoring	1. Create the geofence around child location. 2. Continuously monitoring the childlocation. 3. Notifications send when child cross the geofence and chid face any issues.
FR-3	User requirement	1. Easily upgrade to any environments. 2. Easy to handle. 3. Gives more accuracy. 4. Low power consumption.

4.2 NON-FUNCTIONAL REQUIREMENT

NFR.NO	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Easy to use the gadget.
NFR-2	**Security**	Make sure your child understands that they should always tell you if a stranger approaches, and never to keep this secret
NFR-3	**Reliability**	Child's surroundings can be located with the help of accurate and precise real-

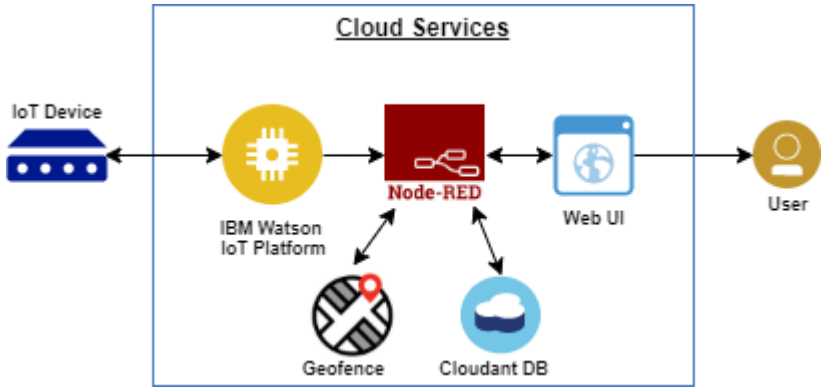
		time location
NFR-4	**Performance**	It easy to track the smart gadget
NFR-5	**Availability**	Audio, video, GPS and sensor
NFR-6	**Scalability**	Young children may refuse to cooperate unless allowed to play with their gadgets





3. PROJECT DESIGN

Data Flow Diagrams:



Technical Architecture

1. Feed the data from the GPS placed in the device to the web interface.
2. The data will display in the web page of the authority(user)
3. The collected data is sent to the data base, where the collected location and predefined geofence location are checked and monitored if the geofence notification send to the parents.

4. The location data is provided to the clouds service and stored
5. The authority monitors the web page continuously to collect the location database the alert to the authority.

Components &technologies

S.No	Components	Description	
1	User Interface	Users had to register and outlook the other device's location Web UI, Mobile App, etc.	HTML, CSS
2	Application Logic-1	Registration of child's and parent's device in each other device.	python
3	Application Logic-2	The child's GPS should be in ON condition, Parent's device should always be correlated to Child's appliance	IBM Watson STT service IBM Watson Assistant
4	Application Logic-3	he child's GPS should be in ON condition, Parent's device should always be correlated to Child's appliance	IBM Watson Assistant IBM Watson STT Service
5	Database	Data Type can be any configuration such as arbitrary binary data, or text. Location history is stored in the cloud and the values include distance, latitude, and longitude. A user-defined blob of data transmitter from Cloud IoT Core to a device	MySQL, NoSQL, SQLite.
6	File Storage	Files will be labelled with what they encompass and how long they should be kept.	IBM Block Storage or Other Storage Service or Local Filesystem
7	Cloud Database	Users install tracking software on a cloud infrastructure to perpetrate the database	IBM DATA BASE
8	External API	The purpose of the external API employed in the device is to exploit the internet for communicating and executing allotted operations efficiently	IBM Weather API, Aadhar API, Geo-Location Lookup
9	Machine Learning Model	IoT and machine learning deliver insights otherwise hidden in data for prompt, automated retorts and enhanced governing.	Object Recognition Model, Danger Prediction Model.
10	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server chassis: Wearable high-tech mechanism. Cloud Server Configuration: a tremendous network that reinforces IoT devices and applications	Local, Cloud Foundry,Underlying Infrastructure.

Application Characersitics:

S.no	Characteristics	Description	Technology
1.	Open source Frameworks	Tracking the location of children	Random data in python script.
2.	Security Implementation	Device ID,IBM cloud andWatson account.	Eg.SHA-256,Encryptions,IAM controls, OWSAP etc.
3.	Scalable Architecture	Upgrade	IBM cloud
4.	Availability	The app contains the Location data of the children.	GPS, Python script
5.	Performance	The system continuously update thelocation data and if the children cross the geofence it will show alert.	Mobile app,Web UI

5.3User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria
Parents	Registration	USN-1	As a parent application registration process through email ,password	I can access my account dashboard
Parents	Login	USN-2	As a parent, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm
Parents	Gadget	USN-3	As a parent to guide a child to use the gadget as safety for child	He ready to use the the application
Parents	Log out	USN-4	As a parent, I can register for the application	When the parent see the child activity in school or parks in some distance to detect the alarm sound to the parents. the gadget OFF mode when the child return to home.

6.PROJECT PLANNING & SCHEDULING
6.1.SPRINT PLANNING & ESTIMATION
PROJECT PLANNING PHASE
PROJECT PLANNING TEMPLATE (PRODUCT BACKLOG,SPRINT PLANNING,STORIES,STORY POINTS)

Sprint	Functional Requirement(Epic)	User Story Number	User Story / Task	Story Points	Priority	Team members
Sprint -1	Login	USN-1	As a customer, I might ensure login credential through gmail ease manner for the purpose of sending alert message to the parents or guardians (or) informing through normal message.	2	High	Subalakshmi, Ranjithkumar, Tamizh, Velmurugan

Sprint -1	Registrati on	USN-2	As a user,I have to registered my details and tools details in a simple and easy manner by considering the safety of child, this registered system sends notification to.	2	High	Subalakshmi, Tamizh, Ranjithkumar, Velmurugan

Sprint-2	Dashboar d	USN-3	As a user, In case of any emergency situation parents(I) must get the alert notification and location of the child.	3	Mediu m	Subalakshmi , Velmurugan
Sprint-3	Dashboar d	USN-4	As a user, I(parent) need to safeguard child and tracking the child's location and it is important to	2	High	Subalakshmi , Ranjithkuma r

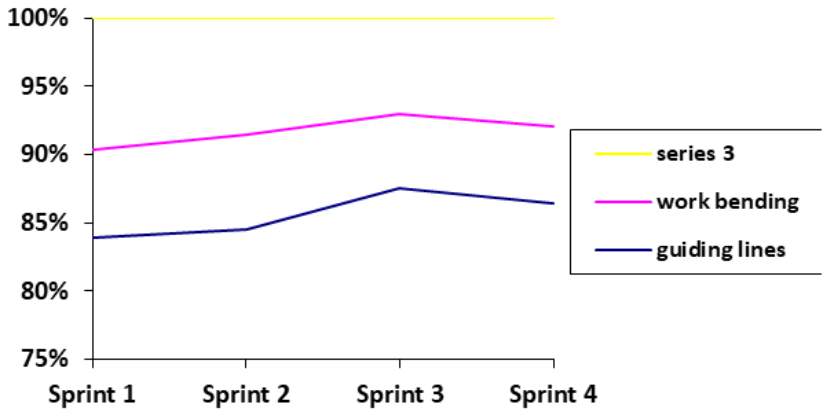
			notify near police station			
Sprint-3	Dashboard	USN-5	As a user, Its good to have a IOT based system to safeguard monitoring without presence of parent.	2	High	Subalakshmi, Tamizh
Sprint -4	Monitoring the environment	USN 1	User can monitor the situation of the environment from a dashboard that displays sensor information about the environment and child health.	2	High	Subalakshmi, Velmurugan
Sprint-4	Event Notification	USN 6	Sending an alert SMS to the parents and guardians in case of panic situation.	2	High	Subalakshmi

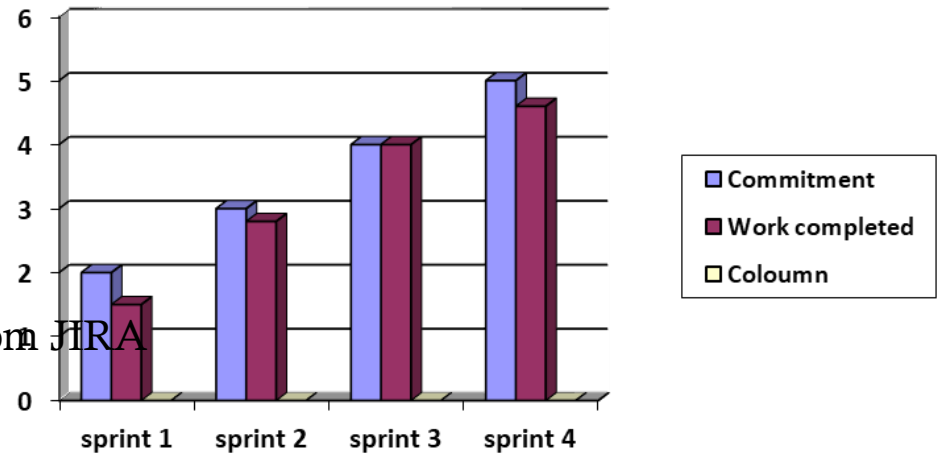
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	4	6 Days	24 Oct 2022	29 oct 2022	4	29 Oct 2022
Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-2	3	6 Days	31 Oct 2022	05 Nov 2022	3	05 Nov 2022
Sprint-3	4	6 Days	07 Nov 2022	12 Nov 2022	4	12 Nov 2022
Sprint-4	4	6 Days	14 Nov 2022	19 Nov 2022	4	19 Nov 2022

Velocity:

Imagine we have a10-day sprint duration, and the velocity of the team 20(points per sprint).Let's calculate the team's average velocity (AV) iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$





6.3 Report From JIRA

7.CODING AND SOLUTIONING:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "1dsau7"
deviceType = "child"
deviceId = "2502"
authMethod = "token"
authToken = "234567890"
#api key {a-illza1-mbdxqo6z0s}
#api token {zSYzISuAWF&F_x7GkT}


try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
method":authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #……................................
except Exception as e:
   print("Caught exception connecting device: %s" % str(e))
   sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of
type "greeting" 10 times
print("power on ")

print("checking connection to waston iot…")
time.sleep(2)
deviceCli.connect()
print("dear user … welcome to IBM-IOT ")
print("i can provide your children live location and temperature ")
name=str(input("enter your child name:"))
while True:



        temperature=random.randint(20,50)#random temperature for your child
        latitude=random.uniform(10.781377,10.78643)#random latitude for your child
        longitude=random.uniform(79.129113,79.134014)#random longitude for your child
        a="Child inside the geofence"
        b=" Child outside the geofence"
```

```
        c="High temperature"
        d="Low temperature"
        x={'your_child_Zone':a}
        y={'your_child_Zone':b}
        z={'temp_condition':c}
        w={'temp_condition':d}




        data = { 'temp' : temperature, 'lat': latitude,'lon':longitude,'name':name }
        #print data
        def myOnPublishCallback():
          print ("Published Temperature = %s C" % temperature, "latitude = %s %%" %
latitude, "longitude = %s %%" % longitude, "to IBM Watson")
          print("\n")
        success = deviceCli.publishEvent("IoTSensorgpsdata", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if latitude>=10.78200 and latitude<=10.786000 and longitude >=79.130000 and
longitude<=79.133000:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=x,qos=0,on_publish=myOnPub
lishCallback)
                print(x)
                print("\n")
        else:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=y,qos=0,on_publish=myOnPub
lishCallback)
                print(y)
                print("\n")

        if (temperature>35):

deviceCli.publishEvent("IoTSensorgpsdata","json",data=z,qos=0,on_publish=myOnPub
lishCallback)
                print(c)
                print("\n")
        else:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=w,qos=0,on_publish=myOnPu
blishCallback)
                print(d)
                print("\n")
```

```
    if not success:
        print("Not connected to IoTF")
        print("\n")
    time.sleep(3)
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

8. TESTING:

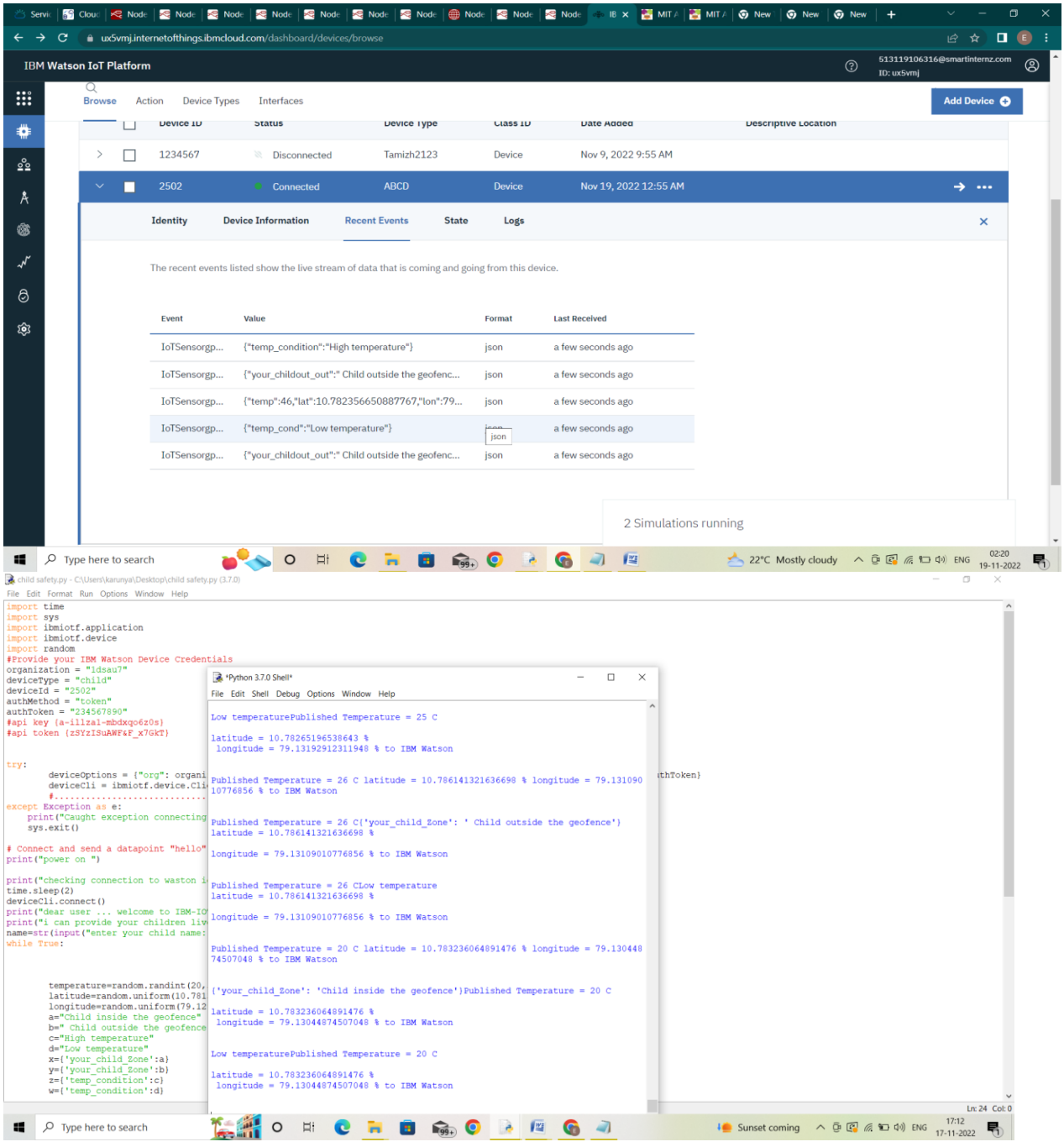
8.1 Test case:

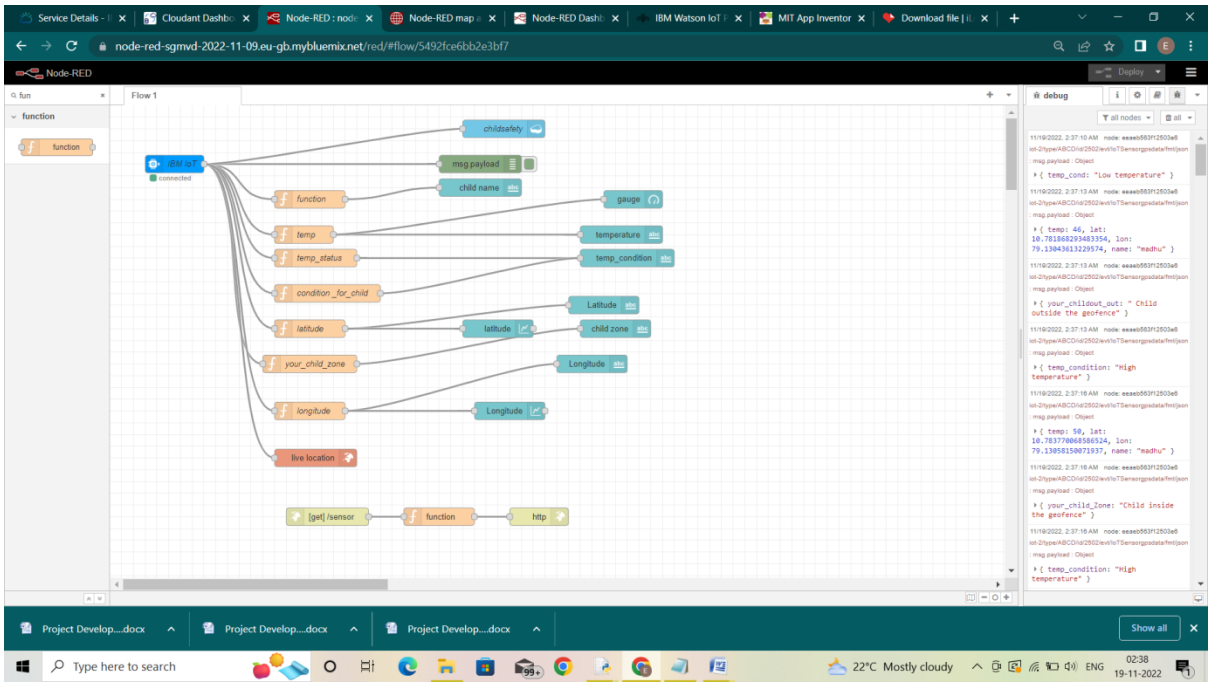
SL.NO	INPUT	OUTPUT	RESULT
01.	Latitude, Longitude Temperature	Inside the geofence, Temperature low	Passed
02.	Latitude, Longitude Temperature	Inside the geofence, Temperature high	Passed
03.	Latitude, Longitude Temperature	Outside the geofence, Temperature low	Passed
04.	Latitude, Longitude Temperature	Outside the geofence, Temperature low	Passed

05.	Latitude, Longitude Temperature	Inside the geofence, Temperature low	Passed
06.	Latitude, Longitude Temperature	Inside the geofence, Temperature high	Passed
07.	Latitude, Longitude Temperature	Outside the geofence,	Passed

		Temperature low	
08.	Latitude, Longitude Temperature	Inside the geofence, Temperature low	Passed
09.	Latitude, Longitude Temperature	Inside the geofence, Temperature high	Passed
10.	Latitude, Longitude Temperature	Inside the geofence, Temperature low	Passed
11.	Latitude, Longitude Temperature	Inside the geofence, Temperature low	Passed
12.	Latitude, Longitude Temperature	Outside the geofence, Temperature low	Passed
13.	Latitude, Longitude Temperature	Inside the geofence, Temperature low	Passed
14.	Latitude, Longitude Temperature	Inside the geofence, Temperature high	Passed
15.	Latitude, Longitude Temperature	Outside the geofence, Temperature high	Passed
16.	Latitude, Longitude Temperature	Outside the geofence, Temperature low	Passed
17.	Latitude, Longitude Temperature	Inside the geofence, Temperature low	Passed

18.	Latitude, Longitude Temperature	Inside the geofence, Temperature high	Passed
19.	Latitude, Longitude Temperature	Outside the geofence, Temperature low	Passed
20.	Latitude, Longitude Temperature	Outside the geofence, Temperature high	Passed
21.	Latitude, Longitude Temperature	Inside the geofence, Temperature low	Passed
22.	Latitude, Longitude Temperature	Inside the geofence, Temperature high	Passed
23.	Latitude, Longitude Temperature	Outside the geofence, Temperature low	Passed
24.	Latitude, Longitude Temperature	Inside the geofence, Temperature low	Passed
25.	Latitude, Longitude Temperature	Outside the geofence, Temperature low	Passed


ux5vmj.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
513119106316@smartinternz.com
ID: ux5vmj
Browse
Action
Device Types
Interfaces
Add Device
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
1234567
Disconnected
Tamizh2123
Device
Nov 9, 2022 9:55 AM
2502
Connected
ABCD
Device
Nov 19, 2022 12:55 AM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
IoTSensorgp...
{"temp_condition":"High temperature"}
json
a few seconds ago
IoTSensorgp...
{"your_childout_out":" Child outside the geofenc...
json
a few seconds ago
IoTSensorgp...
{"temp":46,"lat":10.782356650887767,"lon":79...
json
a few seconds ago
IoTSensorgp...
{"temp_cond":"Low temperature"}
json
a few seconds ago
IoTSensorgp...
{"your_childout_out":" Child outside the geofenc...
json
a few seconds ago
2 Simulations running
Type here to search
22°C Mostly cloudy
ENG
02:20
19-11-2022
child safety.py - C:\Users\karunya\Desktop\child safety.py (3.7.0)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "idsau7"
deviceType = "child"
deviceId = "2502"
authMethod = "token"
authToken = "234567890"
#api key {a-illzal-mbdxqo6z0s}
#api token {zSYzISuAWF&F_x7GkT}
try:
deviceOptions = {"org": organi
deviceCli = ibmiotf.device.Cli
#.....
except Exception as e:
print("Caught exception connecting
sys.exit()
# Connect and send a datapoint "hello"
print("power on ")
print("checking connection to waston i
time.sleep(2)
deviceCli.connect()
print("dear user ... welcome to IBM-IO
print("i can provide your children liv
name=str(input("enter your child name:
while True:
temperature=random.randint(20,
latitude=random.uniform(10.781
longitude=random.uniform(79.12
a="Child inside the geofence"
b=" Child outside the geofence
c="High temperature"
d="Low temperature"
x={'your_child_Zone':a}
y={'your_child_Zone':b}
z={'temp_condition':c}
w={'temp_condition':d}
ithToken}
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Low temperaturePublished Temperature = 25 C
latitude = 10.78265196538643 %
longitude = 79.13192912311948 % to IBM Watson
Published Temperature = 26 C latitude = 10.786141321636698 % longitude = 79.131090
10776856 % to IBM Watson
Published Temperature = 26 C{'your_child_Zone': ' Child outside the geofence'}
latitude = 10.786141321636698 %
longitude = 79.13109010776856 % to IBM Watson
Published Temperature = 26 CLow temperature
latitude = 10.786141321636698 %
longitude = 79.13109010776856 % to IBM Watson
Published Temperature = 20 C latitude = 10.783236064891476 % longitude = 79.130448
74507048 % to IBM Watson
{'your_child_Zone': 'Child inside the geofence'}Published Temperature = 20 C
latitude = 10.783236064891476 %
longitude = 79.13044874507048 % to IBM Watson
Low temperaturePublished Temperature = 20 C
latitude = 10.783236064891476 %
longitude = 79.13044874507048 % to IBM Watson
Ln: 24 Col: 0
Type here to search
Sunset coming
ENG
17:12
17-11-2022



8.2 User Acceptance Testing:

Purpose of Document:

The purpose of this is to briefly explain the test coverage and open issues of the IOT Based Safety Gadget for Child Safety Monitoring and Notification project at the time of the release to User Acceptance Testing(UAT)

Defect Analysis:

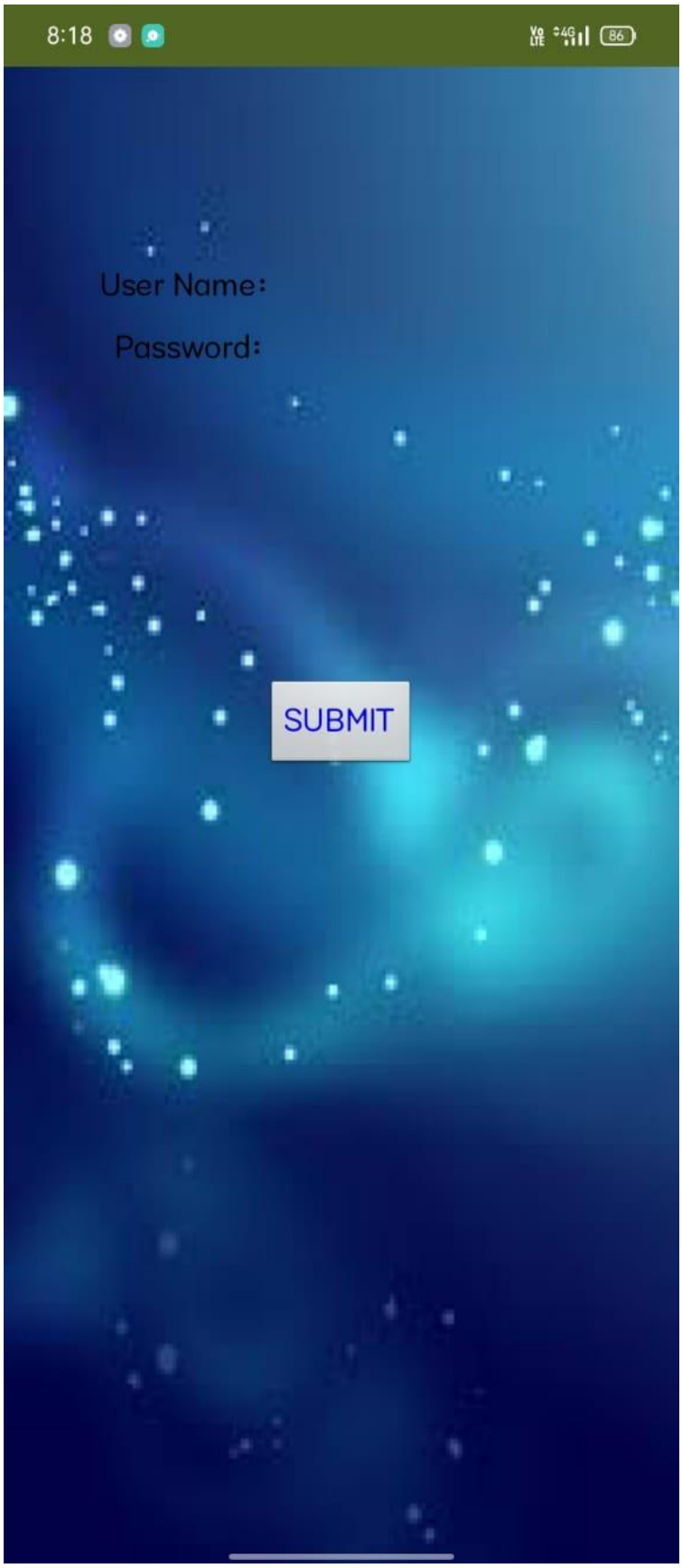
This report shows the numbers or closed bugs at each severity level, and how they were resolved

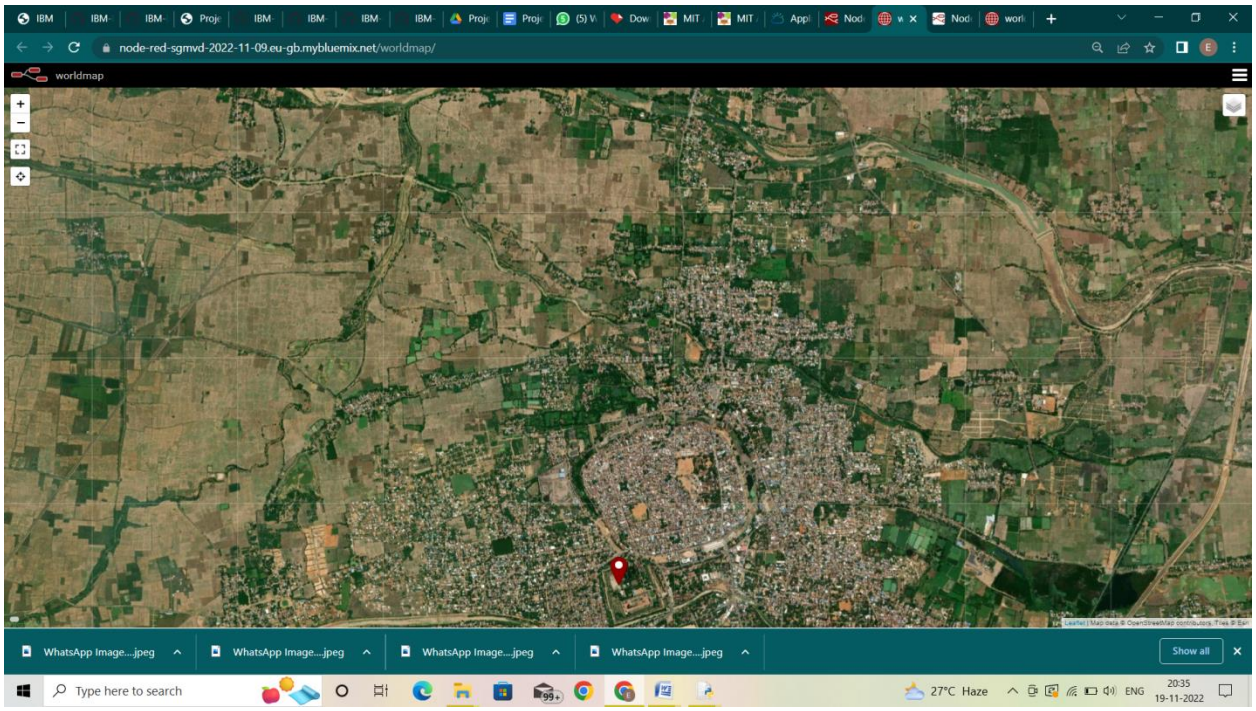
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	3	2	3	13
Duplicate	1	0	0	0	1
External	2	2	0	1	5
Fixed	6	5	3	10	24
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	2	1	3
Totals	14	10	9	16	49

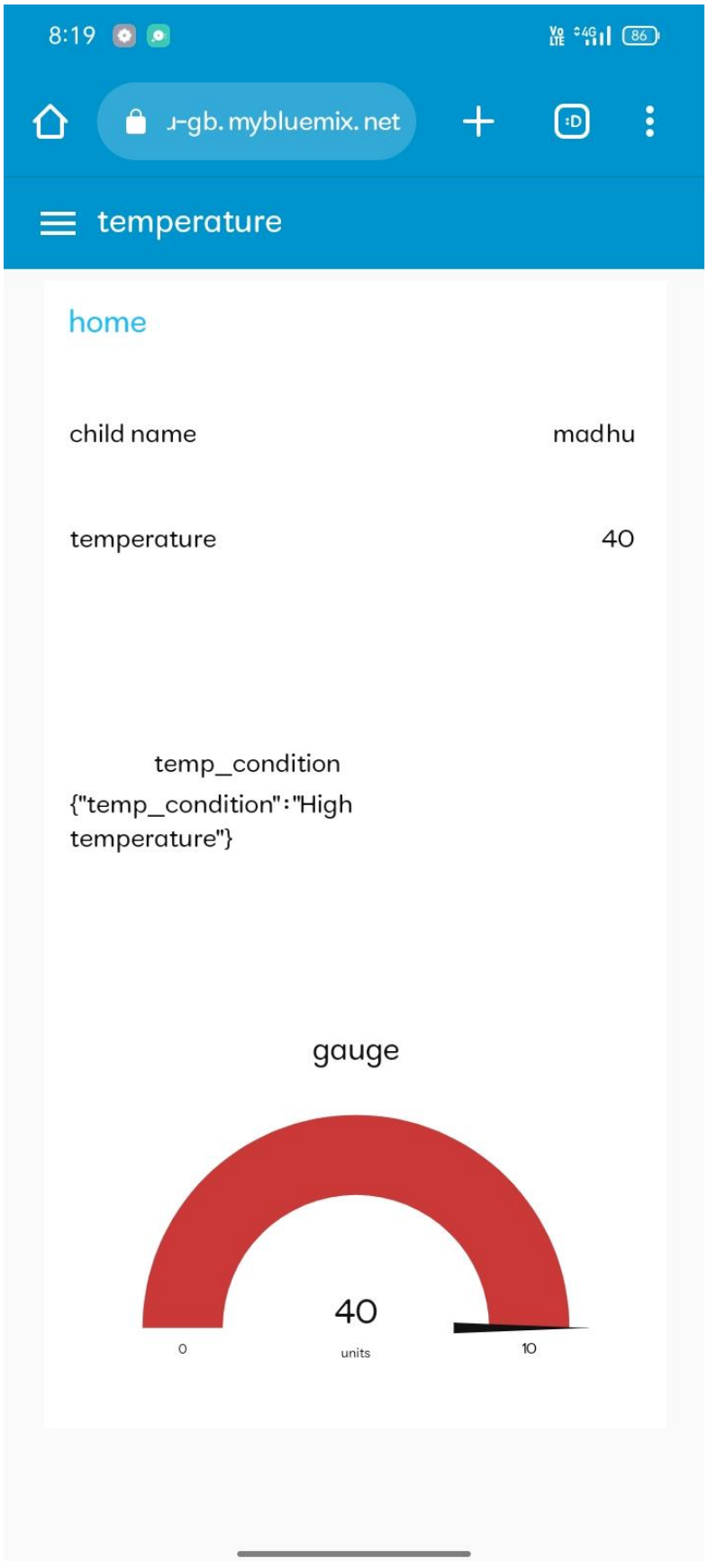
Test Case Analysis:

This report shows the number of the test cases that cases that have passed, failed and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	30	0	0	30
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2


8:18
User Name:
Password:
SUBMIT


node-red-sgmvd-2022-11-09.eu-gb.mybluemix.net/worldmap/
worldmap
WhatsApp Image....jpeg
WhatsApp Image....jpeg
WhatsApp Image....jpeg
WhatsApp Image....jpeg
Show all
Type here to search
27°C Haze
ENG
20:35
19-11-2022


8:19
ɹ-gb. mybluemix. net
temperature
home
child name
madhu
temperature
40
temp_condition
{"temp_condition":"High temperature"}
gauge
40
units
0
10

9. RESULTS:

9.1 Performance metrics

			NFT - Risk Assessment						
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	Safety Gadget for Child Safety Monitoring and Notification	existing	Moderate	Low	Moderate	high initial cost	>5 to 10%	RED	this system or child safety monitoring and notification is already existing
	difficult in creation of geofence					DP 3			They do not Give live location status
									the risk rate is moderate

NFT - Detailed Test Plan				
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Safety Gadget for Child Safety Monitoring and Notification	make a call if the child cross the geofence	Assumptions	

End Of Test Report							
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
	Safety Gadget for Child Safety Monitoring and Notification	issues occurred while running while connecting with the device while creating geofence		got location of the child got some error while connecting with the device difficult in creation of geofence		to determine the frequently update locations	detected and closed	

10. ADVANTAGES AND DISADVANTAGES:

Advantages:

1. Save the life of the children
2. Parent's do their work peacefully with worrying about their children.
3. Continuously monitoring the children.
4. Saves time.
5. Recovery of the children is easy, if the children lost.

Disadvantages;

1. Young children may refuse to unless allowed to play with their gadgets.
2. Easily misusing the device.
3. No water proof.

11. CONCLUSION:

The child tracking system that parents track the movements of children with the help of GPS technology. The entire location data is stored in database. This proposed app can shoes the whether the children inside the geofence or outside the geofence to the parent's mobile. Even if the software is not running the details are shown. It is because location access is available in the background and the software performs well on the mobile device. Based on the availability of the parent user, additional geofences may be required. Performance Requirments are summarized as follows : login ,location status ,temperature ,live on map etc. The system shall allow the user to create and/or log in to account. The system shall allow the user to the current location of the children using GPS.

12. FUTURE SCOPE:

1. Childs surrounding can be located with the help of accurate and precise real time location
2. Surrounding environment temperature , SOS light along with Distress buzzers are provided in this system.
3. If child crosses the geofence, call goes to the registered mobile number's
4. This gadgets will be modified that has been suitable for all environments.

13.APPENDIX:

Python code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "1dsau7"
deviceType = "child"
deviceId = "2502"
authMethod = "token"
authToken = "234567890"
#api key {a-illza1-mbdxqo6z0s}
#api token {zSYzISuAWF&F_x7GkT}


try:
        deviceOptions = {"org": organization, "type": deviceType, "id":
deviceId, "auth-method":authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #.....
except Exception as e:
   print("Caught exception connecting device: %s" % str(e))
   sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud
as an event of type "greeting" 10 times
print("power on ")

print("checking connection to waston iot...")
time.sleep(2)
deviceCli.connect()
print("dear user ... welcome to IBM-IOT ")
print("i can provide your children live location and temperature ")
```

```
name=str(input("enter your child name:"))
while True:



    temperature=random.randint(20,50)#random temperature for your
child
    latitude=random.uniform(10.781377,10.78643)#random latitude for
your child
    longitude=random.uniform(79.129113,79.134014)#random longitude
for your child
    a="Child inside the geofence"
    b=" Child outside the geofence"
    c="High temperature"
    d="Low temperature"
    x={'your_child_Zone':a}
    y={'your_child_Zone':b}
    z={'temp_condition':c}
    w={'temp_condition':d}




    data = { 'temp' : temperature, 'lat':
latitude,'lon':longitude,'name':name }
    #print data
    def myOnPublishCallback():
      print ("Published Temperature = %s C" % temperature, "latitude =
%s %%" % latitude, "longitude = %s %%" % longitude, "to IBM Watson")
      print("\n")
    success = deviceCli.publishEvent("IoTSensorgpsdata", "json", data,
qos=0, on_publish=myOnPublishCallback)
    if latitude>=10.78200 and latitude<=10.786000 and longitude
>=79.130000 and longitude<=79.133000:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=x,qos=0,on_publ
ish=myOnPublishCallback)
        print(x)
        print("\n")
```

```
        else:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=y,qos=0,on_publi
sh=myOnPublishCallback)
                print(y)
                print("\n")

        if (temperature>35):

deviceCli.publishEvent("IoTSensorgpsdata","json",data=z,qos=0,on_publi
sh=myOnPublishCallback)
                print(c)
                print("\n")
        else:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=w,qos=0,on_pub
lish=myOnPublishCallback)
                print(d)
                print("\n")
        if not success:
                print("Not connected to IoTF")
                print("\n")
        time.sleep(3)
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

GitHub & Project demo link:
Github link: https://github.com/IBM-EPBL/IBM-Project-14883-1659591535