http://doi.org/10.29194/NJES.22030172

Design and Implementation of Children Tracking System using IoT

Sama Salam Samaan[1]

Author affiliation:
1) Computer Engineering Department, University of Technology, Baghdad-Iraq.
samasamaan@gmail.com

Paper History:

Received: 17th March 2019

Revised: 23th April 2019

Accepted: 17th Sep. 2019

Abstract

In the past few years, all over the world, crime against children has been on the rise, and parents always worry about their children whenever they are outside. For this reason, tracking and monitoring children have become a considerable necessity. This paper presents an outdoor IoT tracking system which consists of a child module and a parent module. The child module monitors the child location in real time and sends the information to a database in the cloud which forwards it to the parent module (represented as a mobile application). This information is shown in the application as a location on Google maps. The mobile application is designed for this purpose in addition to a number of extra functions. A Raspberry Pi Zero Wireless is used with a GSM/GPS module on shield to provide mobile communication, internet and to determine location. Implementation results for the suggested system are provided which shows that when the child leaves a pre-set safe area, a warring message pops up on the parent's mobile and a path from the current parent location to the child location is shown on a map.

Keywords: IoT, Database on the Cloud, Firebase DB, Raspberry Pi Z W, GSM module, GPS module, Python, Android Application.

تصميم وتنفيذ نظام مراقبة للاطفال بأستخدام انترنت الاشياء

سما سلام سمعان

الخلاصة:

في السنوات القليلة السابقة، شكلت الجرائم بحق الاطفال ارتفاعا ملحوظا في مناطق مختلفة من العالم مما يشكل سبباً لقلق الوالدين كلما كانو خارج المنزل مع الاطفال. لهذا السبب، أصبحت مراقبة وتعقب الاطفال ذات اهمية كبيرة. هذا العمل يقدم نظام تعقب للاطفال بأستخدام انترنت الاشياء. النظام المقترح يتكون من جزئين، الاول يكون مع الطفل والجزء الثاني مع الوالدين. الجزء الخاص بالطفل يراقب موقع الطفل بالزمن الحقيقي ويقوم بأرسال هذه المعلومات الى قاعدة بيانات في السُحُب والتي تقوم بأرسالها الى الوالدين (الجزء الثاني) والمتمثل بتطبيق مثبت على الهاتف الذكي. هذه المعلومات تظهر كَموقع على تطبيق خارطة كوكل. تم تصميم تطبيق للهاتف الذكي والذي يوفر مجموعة اضافية من الوظائق فضلاً عما هو مذكور انفاً. تم استخدام Raspberry Pi Zero Wireless مع GSM/GPS لتوفير امكانية الاتصال المتنقل والانترنت ولتحديد الموقع. اظهرت نتائج التنفيذ للنظام المقترح انه عندما يصبح الطفل خارج المنطقة الامنة والمحددة سابقاً من قبل الوالدين، فأن رسالة تنبيه سترسل الى هاتف الوالدين كما سيظهر على خارطة كوكل المسار الواجب سلكه من موقع الوالدين الحالي الى موقع الطفل.

1. Introduction

Children tracking system is used to assure parents that their children are safe from suspicious actions or loss (J. Mohan, 2019). The information related to the child location is sent to a database in the cloud in real time and this location is sent directly to an application which is set up in the parent's mobile and this location is shown on Google map.

Compared to Arduino, Raspberry Pi connects easily to the internet. Arduino is not as powerful as the Raspberry Pi and it can only be programmed using Arduino or C/C++ while the Pi has the entire Linux software stack available and it can be programmed using a variety of languages.

To provide an overview of earlier works, some researches presented by diverse authors are reviewed:

- (Shatha K. Jawad et al., 2009) they designed a system to work indoor or outdoor. GPS is used for outdoor distance computation, while for indoor distance computation; the RF signal amplitude alteration is used.

- (Aswathi Dileep et al., 2015) includes a campus, a child module and a parent module. When the child is out of the campus, the child module will send an alert message to the parent module. A microcontroller ATMEGA 162 is used with GSM and GPS module which helps to track the child movements.
- (P. SANTHA RAJ and V. ANURADHA, 2016) they designed a system which contains a child module and two receiver modules for obtaining the information about the missing child on periodical basis. ARM7 microcontroller, GPS, GSM and Voice circuit is contained in the child module and the receiver module includes an Android mobile phone application for the parent and the other as monitoring database in the school control room.
- (Mustafa Saeed et al., 2016) they used fingerprint verification as tool for children identification. The proposed technique sends SMS though HTTP to the parent.

2. System Design

This section describes the conceptual design of the Children Tracking System as shown in Fig.1. The system consists of two parts, the child module and the parent module. The child module consists of a Raspberry Pi Zero - Wireless with a GSM/GPS module. The parent module is represented by an application running on an android mobile. The child movement is tracked by the GPS module and the location is sent in real time to Firebase database on the cloud which forwards it to an application set up on the parent's mobile phone. All this is done via Internet using 3G service which is available in the GSM module. The Firebase Real-time Database is a NoSQL, cloud-hosted database. Data is stored as JSON, synchronized and distributed in real-time to each connected client. When an app is built, all of the clients will share one Real-time Database instance and automatically receive updates of the newest data. Data remains available when the app goes offline. It is free, easy to use and implement.

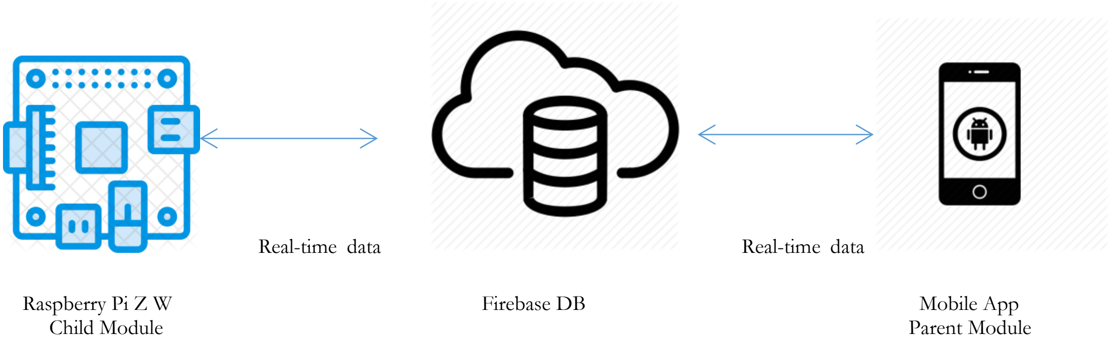


Figure (1): Conceptual Design of the Children Tracking System

3. The Child Module

3.1 The Hardware Part

The child module shown in Fig.2 consists of the following hardware parts:

- Raspberry Pi Zero - Wireless

The Raspberry Pi is a common Single Board Computer (SBC). It is a complete minicomputer packed into a single board. The introduction of the Raspberry Pi Zero permits the embedding of an entire computer in smaller projects like the one proposed in this paper. The Raspberry Pi Zero W has all the functionality of the original Pi Zero, but comes with added GSM/GPS Raspberry pi modular. The Pi must be able to use both the GPS and GSM simultaneously. The modular should be connected to the pi with no USBs. The Raspberry pi hat/modular receives AT-COMANDS from the raspberry pi through IO pins and responds with the appropriate data (a map location, SMS messages, or call events). The extension board shown in Fig.2 (b) is simply plugged onto the GPIOs of the Raspberry Pi. The RPI Expansion has a SIM card holder.

- SIM Card

We can connect our Pi to the internet through Ethernet or Wi-Fi.

- SD card

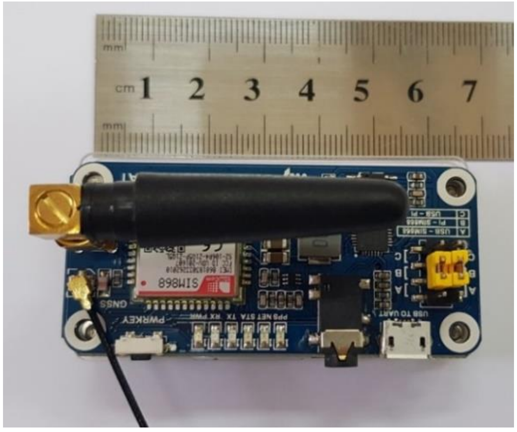
The device's main storage which contains the OS files (image file), the uploaded code and a log of the device's routes.

- External microphone and Speaker

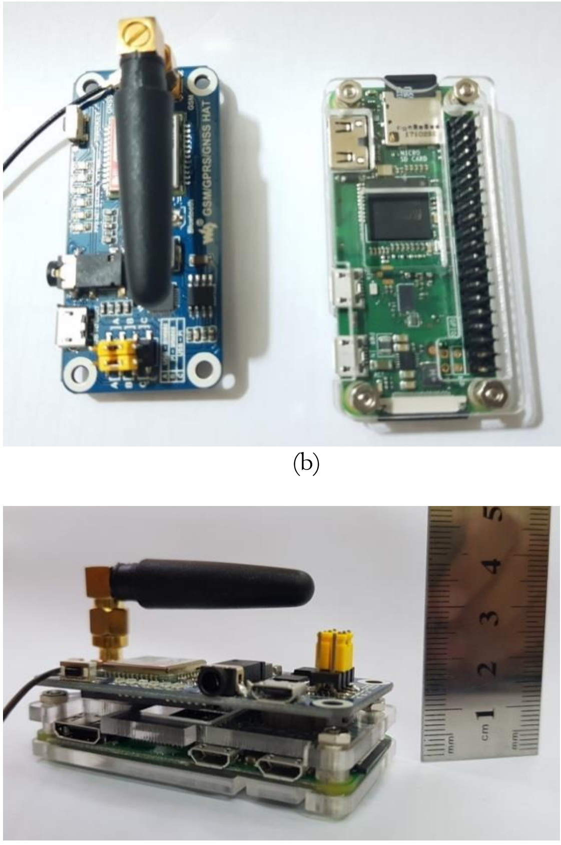
Used to initiate calls with the device or just to hear the device surroundings.

- Battery and Charger

With many options for outdoors and portable Raspberry Pi projects, it's important to have a flexible power solution which helps the Raspberry Pi to run outdoors, whatever the project. In this project we use a power bank (5V).



(a)



(b)

(c)

Figure (2): The Raspberry Pi Z W Child Module

3.2 The Software Part

The child module has the following software parts:

- **Raspberry Pi Zero W Operating System**

The preferred choice for running the Raspberry Pi Zero W as a computer is Raspbian, which is the Pi's official operating system. Raspbian, which is Linux based, is a custom-version of Debian that has been optimized to run on the Pi's hardware. It is used to run the uploaded code and send/receive commands and data to the GSM/GPS modular through GPIO pins.

- **The uploaded code**

The code is uploaded into the Raspberry pi SD card. This code is written in Python. It receives the child location from the GPS modular (longitude and latitude) and forwards it directly to the Firebase database.

The reason behind using Python is that it is an essential programming language hosted on the Raspberry Pi. It is a robust language that's easy to use and it allows the binding of the project to the real world. A Flowchart that describes how the system is working is shown in Fig.3.

We divide the code into serval parts that run through a cycle, as follow:

- Cycle Part
- GPS Part
- PPP Part
- Firebase Part
- SMS Part

3.2.1 Cycle Part

It is a shell script (bash file) which contains the essential Linux commands to run the Python codes in specific time slots and to manage when to turn on internet connection and when to turn it off. This cycle was made duo the lack of multiple physical serial ports in Raspberry Pi; since it has only one that can run only one task at a time. The internet connection through 2G cellular networks on the GSM board uses the serial port to transfer data from Raspberry Pi to the GSM. On the other hand, to make use of the GPS, it must communicate with the Raspberry Pi through the serial port, so now we have a problem of the usage of that port. As a result, we divide time between these two processes; first we use the GPS to get coordinates and save them on the device storage in a shared file that can be accessed by other parties, then turn the GPS off, start the PPP (Point-to-Point Protocol) to establish a direct connection between the Raspberry Pi and a cellular tower in order to have an internet connection. If the connection is established, we start running the firebase code to send the data to the firebase database. After sending the data, the PPP connection is turned off, and the GPS code is started again to get new coordinates. If we couldn't establish the internet connection, the SMS part starts by sending the data through an SMS message directly to the phone. It takes approximately 3 minutes to finish one cycle. The cycle is shown in the Fig.4.

3.2.2 GPS Part

Using Python to write the GPS code, the Raspberry Pi communicates with the GPS modular by AT commands. First, we initialize the modular to start reading coordinates and send them through the serial port to Raspberry Pi. It sends the latitude, longitude and altitude of the Raspberry Pi. After receiving the data, if it is valid, the data will be saved in a shared file on the storage of the Raspberry Pi; in our case it is the SD card. The code will run for a fixed period of time (in our case for 60 seconds) to ensure the data reading operation is completed. After that, the GPS code exits, and the cycle continues to the next Part. Another job for this part is when receiving the data string, it would also include the SIM card status, if there is any received call, it will be detected here. We made the device answer the calls only if they were from the parents' phone numbers.

3.2.3 PPP Part

Point-to-Point Protocol (PPP) is a data link layer (layer 2) communications protocol used to establish a direct connection between two nodes. In this work, we use Linux as OS so it's easy to setup and configure the OS to use PPP as the main traffic gate for all the packets that leave the Raspberry Pi. After that, we have two simple commands to turn it on and off, we put them in the cycle code, it's the one that controls the connection as we mentioned before.

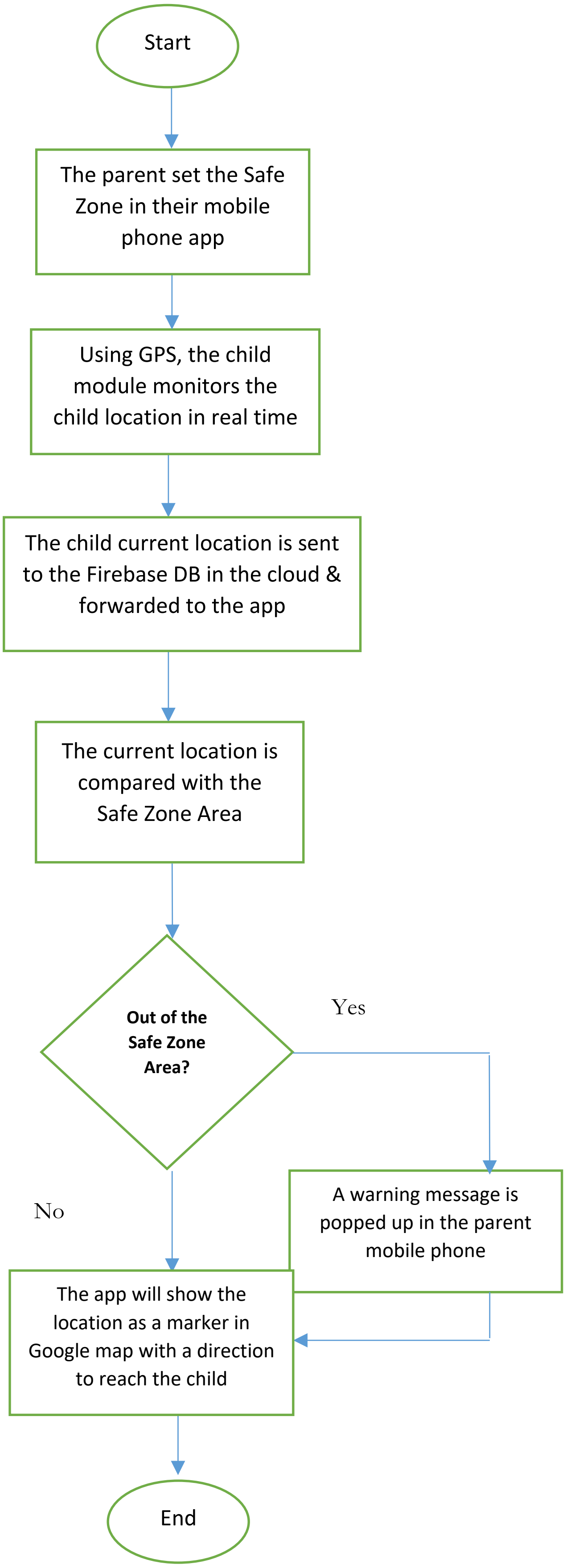

Start
The parent set the Safe Zone in their mobile phone app
Using GPS, the child module monitors the child location in real time
The child current location is sent to the Firebase DB in the cloud & forwarded to the app
The current location is compared with the Safe Zone Area
Out of the Safe Zone Area?
Yes
No
A warning message is popped up in the parent mobile phone
The app will show the location as a marker in Google map with a direction to reach the child
End

Figure (3): Flowchart that describes how the system is working

3.2.4 Firebase Part

This part of Raspberry Pi code is responsible of reading the data from the shared file, then establishing a connection with the firebase database and uploading the data to store it there. In order to establish the connection with the firebase we need a database URL, API Key, authentication Domain and storage Bucket, all these essential info are supplied in the code with config variable, then we use a library named Pyrebase to manage the connection and the process of uploading the data. After the uploading of data is done, the code exits, and the cycle continue to the next part.

3.2.5 SMS Part

SMS part is created as a backup plan if there is a failure in the process of connecting the device to the internet. It sends the data only if the device is outside the safe zone. In an earlier stage, it receives the coordinates and the radius of the safe zone from the firebase, so it can calculate the distance between the current location of the device and the safe zone and tells if it's inside or outside the zone. Also, it saves the last known location of the device and sends it if the GPS signal is lost within a specific period. In our scenario it is set to for 2 minutes. This feature can be enabled or disabled by the parent from the mobile application, in addition to setting the timeout period.

4. The Parent Module

We developed an android application to represent the end user interface. It contains the essential elements to monitor and control the device, register child info, and set the safe zone location and radius. We will take a deeper look on the app activities. Each activity in the app represents an individual interface layout. The activities are as follows:

- Register device activity
- Enter child info activity
- Dashboard activity
- Map activity
- Control activity

We use Kotlin as the programing language and XML to design the interfaces' layouts. All of that is done using the Android Studio IDE.

4.1 Register device activity

It's the first activity that shows up to the user when it's the first they run the application. The user enters the device ID and the phone number that is associated with the device, it checks the device ID, if it is a valid device, it continues to the next activity (the child info activity). Each device has its own unique ID, to distinguish it between the devices in the database. If the device's ID was already entered by another user (parent), the user is asked if they want to retrieve the child data from the firebase database, if they answered with "yes", then it would go to the dashboard activity, if "no", it would go to the child info activity. This activity layout is shown in Fig.5.

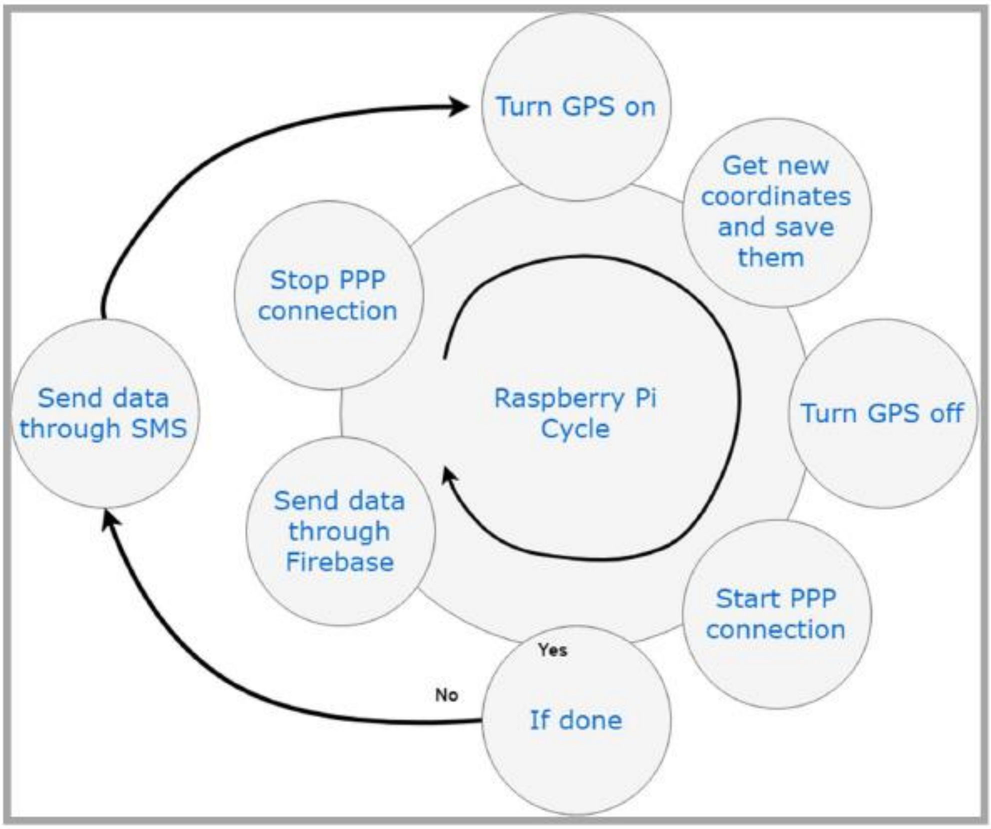


Figure (4): Raspberry Pi Code Cycle

4.2 Enter child info activity

In this activity the user enters the child's information, like the name, age and gender of the child. After the data is entered, it is saved into two locations: the first location is the phone memory (local), and the second location is the firebase database (cloud), so that if there is a second user (the other parent) they will have the same data straight from the firebase database without the need to re-enter the data. The activity layout is shown in Fig.6.

4.3 Dashboard activity

This is the activity that contains the main elements to control the device functions; it also shows the child info in the top section of the layout. The main functions it provides are:

- Shows the child info
- Add new child – button
- Show next child info – button
- Make a call to the device it is associated with – button
- Send alarm – button
- Map – button
- Control – button
- Battery percentage

As a start, this activity (in the background) starts establishing a connection with the Firebase database, and starts listening to data changes. While the device is sending coordinates to the database, the applications receives these coordinates, starts measuring the distance between the device and the safe zone coordinates (safe zone will be discussed in map activity). If the device is inside the safe zone, nothing will happen, but if it is outside the safe zone, a notification will be sent to the phone. If the device is still outside, with each data change the notification will refresh itself. This process will keep working in the background even if the app isn't running in the foreground. The "add new child" button will navigate the user to Enter info activity to add a new child. The "Make call" button is responsible of making a call between the parent's phone and the device. The parent can also mute the mic while calling so they can listen to the surroundings of the child without gaining the attention of anyone who is near the child. Because of Raspberry Pi's lack for a second serial port (as we mentioned in Cycle part), the call will be detected and answered only in the GPS part. The "Send alarm" button was created so that if the device was lost within a short range or inside a building (like when you can't find a remote controller), and its GPS coordinates can't show its specific location, so when the device receives this action it starts making a noise so that the owner can find it. The noise will stop after the user pushes a physical button on the device indicating that they found it. The "Battery" icon shows the battery percentage of the device, so if the battery goes low, it notifies the user to charge the device. The Dashboard activity is shown below in Fig.7.

4.4 Map activity

This activity is responsible for showing the device's location on the map; we used Google maps services for this purpose. The user can set a safe zone simply by clicking on any location on the map and a red circle will be drawn on it, there is also a bar on the top to specify the radius of the safe zone circle. Once that is done, the circle will be saved on the device memory (local), and on the Firebase database (cloud), so that the other users that are connected to the same device can receive the safe zone changes. Let's consider the following scenario: the child goes to school, the parent changes the safe zone from home area to school area, if the other parent didn't know about this change the application will consider the child is out the safe zone (home area). Because of that we made the safe zone change on both devices in real time.

The map activity gives us a new option to consider, we called it nearby area (yellow circle). When you and your child are out, you don't have to change the safe zone each time to enter a new area; the app will keep track of the phone location and calculate the device location for you. If the device is within 10 meters away from the phone, even if it is outside the safe zone (red circle), the warning notification will not be sent.

The map activity layout consists of a few buttons to make it easy to track each marker on the map, they are:

- Mobile loc. To zoom at the mobile's current location.
- Zone loc. To zoom at the safe zone (red circle).
- Tracker loc. To zoom at the device's current or last location.
- Get route is created to draw a path between the mobile's current location and to the tracker's (device) location, to make it easy to track the child.

To distinguish between the current location of the device and its last known location, we made a status indicator label at the top of the map, it would show "Live" if the device is sending its location and "Offline" if the device stops sending its data in a few minutes (specified period). The map activity layout is shown in Fig.8 to Fig.11.

4.5 Control activity

Also called the setting activity, is created to set a number of values in the system, for instance, the timeout of SMS message, enabling and disabling the system and setting the battery warning threshold.

4.6 Firebase Database

Firebase is a Backend-as-a-Service — BaaS — which developed into a next-generation app-development platform on Google Cloud Platform. It's a Real-time Database. When the device uploads a new data message to the firebase database, it forwards this data to the application instantly, the data message form is shown in Fig.12. The first three values represent the time of sending the message, and the last two values represent the latitude and the longitude. The safe zone structure shown in Fig.13 and the child info is stored as shown in Fig.14.

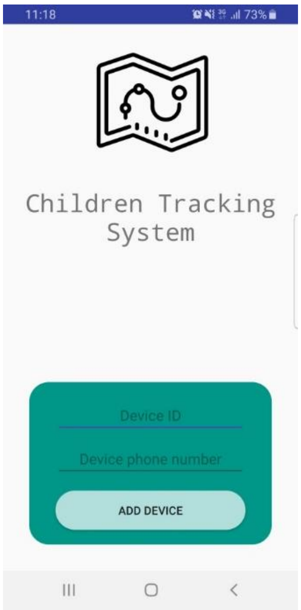


Figure (4): Register Device Activity

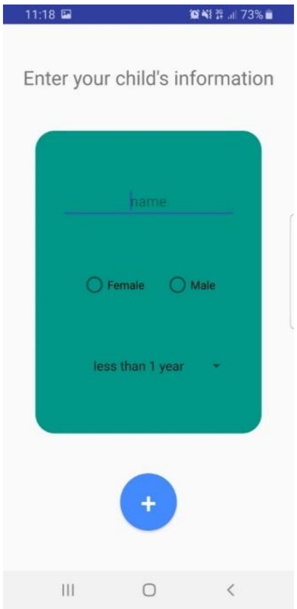


Figure (5): Enter Child Info Activity

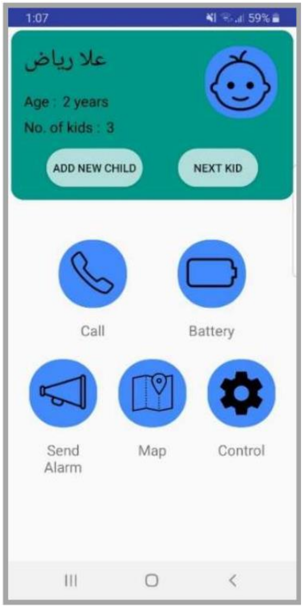
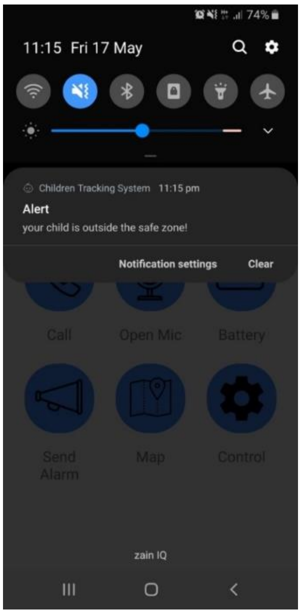


Figure (6): Dashboard Activity



Children Tracking System 11:15 pm
Alert
your child is outside the safe zone!

Figure (7): Child Notification Alert

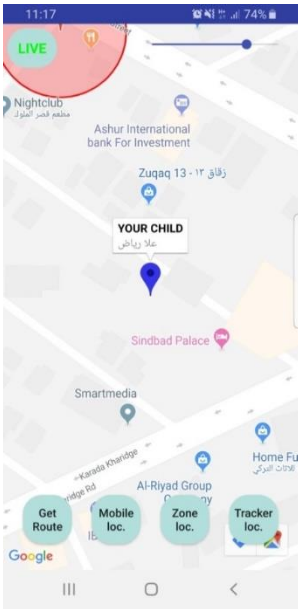


Figure (9): Child Out of Safe Zone Area

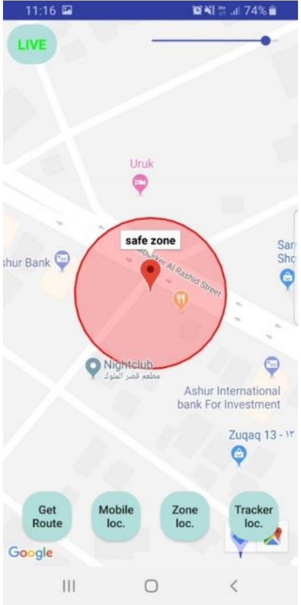


Figure (8): Identifying Safe Zone Area

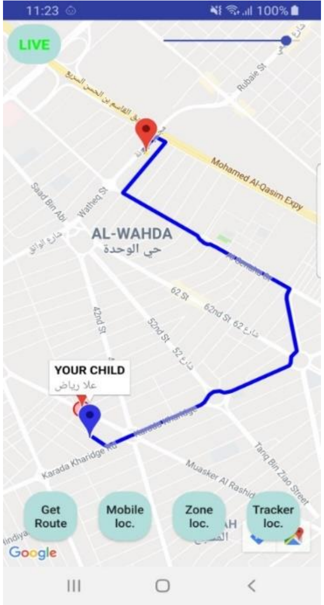
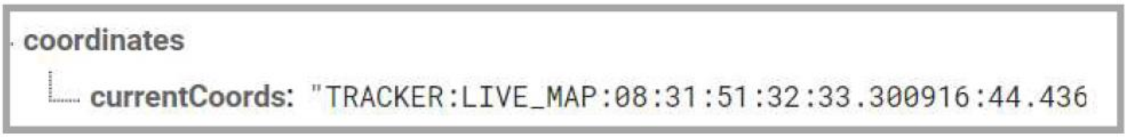
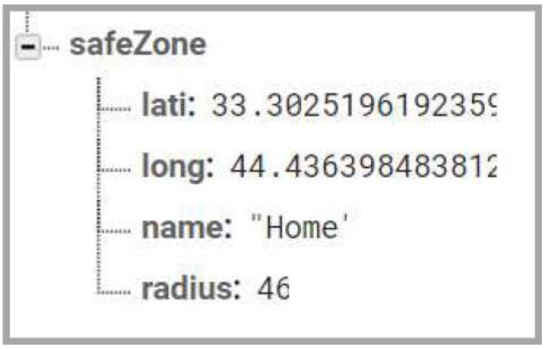


Figure (11): Path from Parent to Child



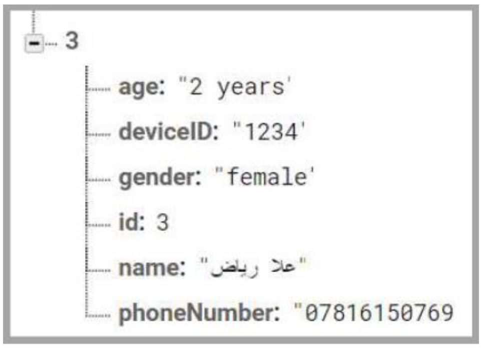
```
coordinates
  currentCoords: "TRACKER:LIVE_MAP:08:31:51:32:33.300916:44.436
```

Figure (12): The Data Message Form



```
safeZone
  lati: 33.302519619235
  long: 44.436398483812
  name: "Home"
  radius: 46
```

Figure (13): The Safe Zone Structure



```
3
  age: "2 years"
  deviceID: "1234"
  gender: "female"
  id: 3
  name: "علا رياض"
  phoneNumber: "07816150769
```

Figure (14): The Child Info

5. Conclusions

In this paper, an IoT children tracking system is developed. The implementation originally focuses on monitoring a child and sending its location to the Firebase database which forwards this information in real time to an application which is created for this purpose and is set up in the parent mobile phone. The parent will be notified if the child has moved outside a specific area (Safe Zone Area). In addition to tracking the child, the app provides a number of services. The designed system allows the parent to monitor multiple children at the same time. This research shows that GSM/GPS tracking technology is a practical approach for monitoring and tracking children with the help of Google database service, Firebase. Except the Internet fee, there is no cost associated with the presented service.

6. References

1. Shatha K. Jawad, Al-Gawagzeh Mohammed Yousef and Balkiest Essa Al-Shagoor, "A Multipurpose Child Tracking System Design and Implementation", International Journal of Soft Computing Applications, ISSN: 1453-2277 Issue 4 (2009), pp.57-68.
2. P. Santha Raj and V. Anuradha, "Design and Implementation of Children Tracking System using ARM7 on Android Mobile Terminals", international Journal of Scientific Engineering and Technology Research, ISSN 2319-8885, Vol.03, Issue.21, September-2014, Pages:4327-4332.
3. Loganathan.M, Aswathi Dileep and Kamatchi.K, "Child Tracking System Based on GSM ", International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), Volume 4, Special Issue 4, April 2015.
4. Rita H. Pawade and Arun N. Gaikwad, "Android Based Children Tracking System", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 6, June 2015.
5. Shubhangi P. Mankar, Monali Pawar and Manisha Shinde, "Child Tracking System based on GPS System", International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), Volume 4, Issue 4, April 2016.
6. A. Saranya, C. Venkatesh and S. Selva Kumar, "Design and Implementation of Automatic Child Monitoring (ACM) System Using Wireless Network", International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 5, Issue. 4, April 2016, pg.356 – 363.
7. Maria Clenisha, Sandra, A Pio Sajin and B Baron Sam, "Survey on Automated Child Monitoring System Using Image Processing", International Journal of Pure and Applied Mathematics, Volume 118, No. 24, 2018.
8. J. Mohan, B.Ragul, M. Saravanakumar, Mrs. V. Hemalatha and Dr. P. Gomathi, "Enhancement Of Smart Child Monitoring System", International Research Journal of Engineering and Technology (IRJET), Volume 06, Issue 02, Feb 2019.