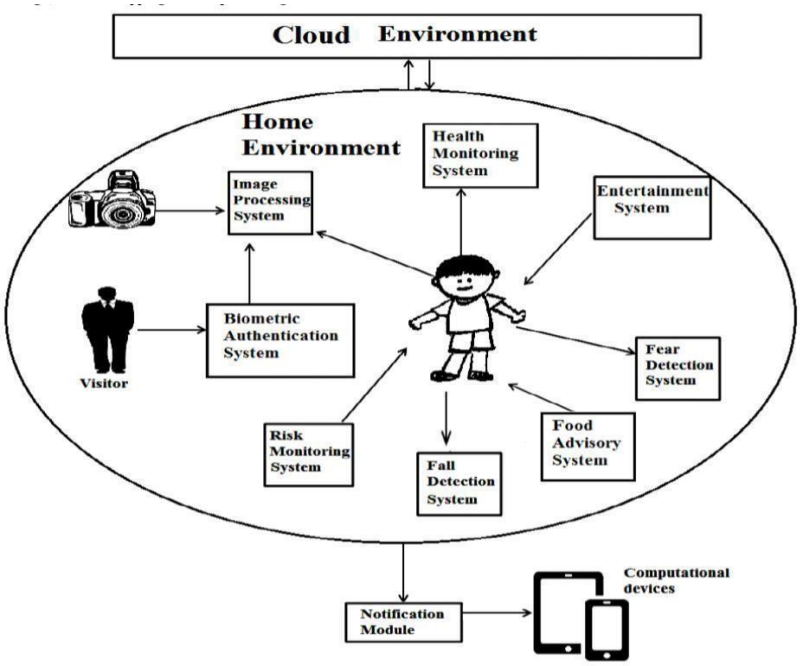
Project Development Delivery Of Sprint 3

DATE	10-Nov-2022
TEAM ID	PNT2022TMID04763
PROJECT NAME	IoT Based Safety Gadget for Child Safety Monitoring & Notification

Thus, the Smart Mom architecture facilitates their work and aids in child care. Additionally, it is presumable that this technique is beneficial for kids between the ages of five and fifteen. Since children under the age of five require special care and children over the age of fifteen can generally be cared for by their mothers, children under the age of five should not be handled by an automated system.The cloud environment and the home environment are the two domains into which the Smart Mom architecture is divided. The number of modules within each domain varies depending on the application system.

The notification is in charge of notifying the computing devices within or outside of the home. Depending on the required use, the computing equipment may be wired or wireless and belong to the child, the governess, the doctor, or the child's mother.



Python Serial Loopback Test

```
import serial

#####Global     Variables#######################################
#be sure to declare the variable as 'global var' in the fxnser = 0

#####FUNCTIONS###################################################
#initialize serial
connectiondefinit_serial():
        COMNUM = 9 #set you COM port # here
        global ser #must be declared in each fxn
        usedser =serial.Serial()
        ser.baudrate = 9600
        ser.port = COMNUM - 1 #starts at 0, so subtract
        1#ser.port ='/dev/ttyUSB0' #uncomment for linux

        #you must specify a timeout (in seconds) so that the# serial port
        doesn'thang
        ser.timeout = 1
        ser.open() #open the serial port

        # print port open or
        closedifser.isOpen():
                print 'Open: ' + ser.portstr
#####SETUP########################################################
#this is a good spot to run your initializationsinit_serial()

#####MAIN     LOOP###############################################
while 1:
        #prints what is sent in on the serial port

        temp = raw_input('Type what you want to send, hit enter:\n\r')
```

```
    ser.write(temp) #write to the serial port
    bytes = ser.readline() #reads in bytes followed by a newline
    print 'You sent: ' + bytes #print to the console
    break #jump out of loop
#hit ctr-c to close python window
```

```
#adjust these values based on your location and m
TRX = -105.1621     #top right longitude
TRY = 40.0868       #top right latitude
BLX = -105.2898     #bottom left longitude
BLY = 40.0010       #bottom left latitude
```

Run the program by typing:

The development of encircling systems has long continued to use high-level language programme design. However, assembly programming still dominates, particularly for systems using digital signal processors (DSP). DSPs are automatic frequency systems written in assembly language by computer operators who are intimately familiar with processor architecture. Performance serves as the primary motivator for this strategy, despite the drawbacks of assembly software architecture when combined with high-level programming.