Final Deliverable Phase

Final Code

Date	19 November 22
Team ID	PNT2022TMID41842
Project Name	IoT based safety gadget for child safety monitoring and notification
Team Size	4

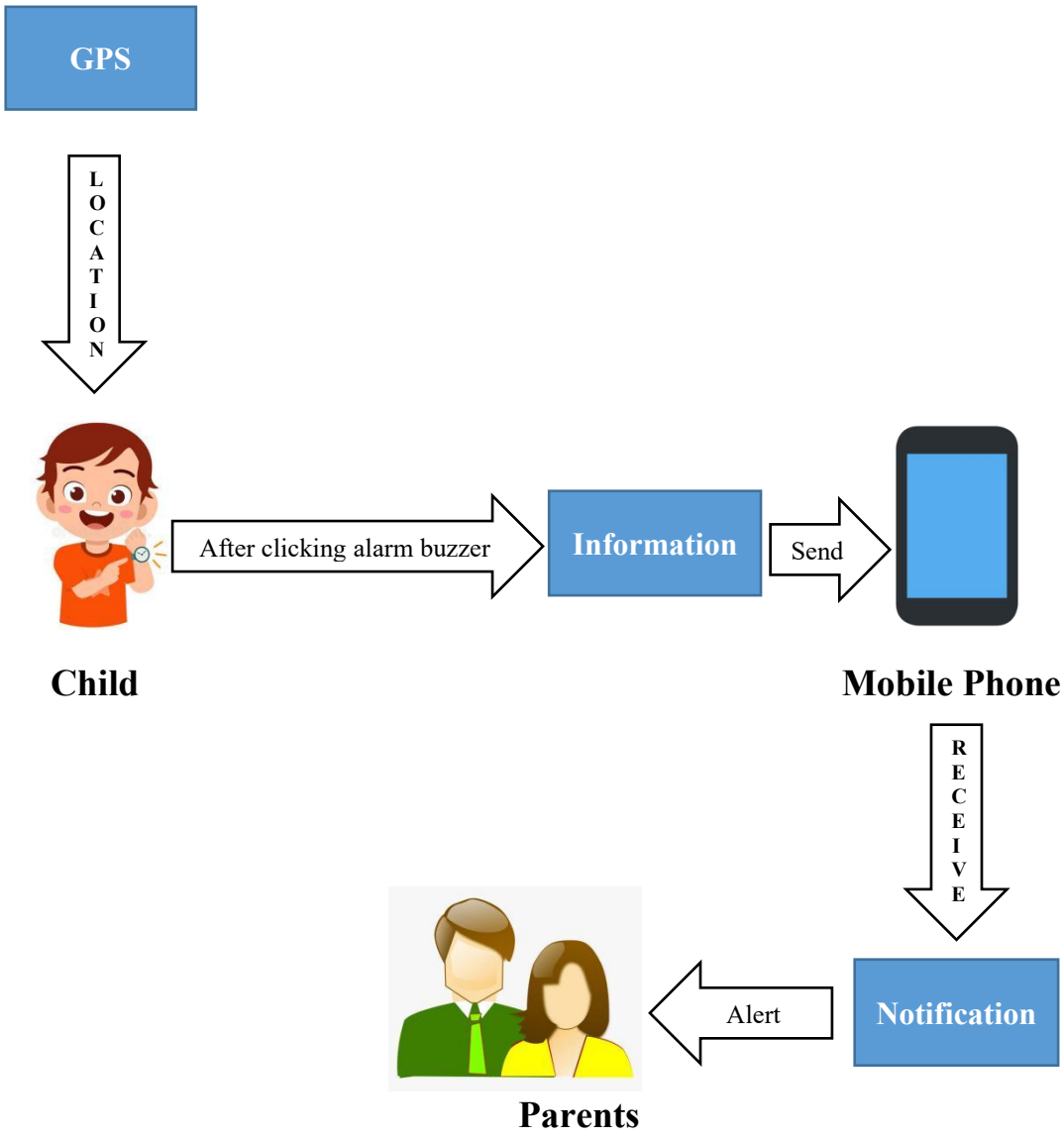
Project coding :

```
import json
import pycountry
import time
from tkinter import Tk, Label, Button, Entry
from phone_iso3166.country import phone_country
from time import sleep
from sinchsms import SinchSMS

class Location_Tracker:
    def _init_(self, App):
        self.window = App
        self.window.title("Phone number Tracker")
        self.window.geometry("500x400")
        self.window.configure(bg="#3f5efb")
        self.window.resizable(False, False)

        #____Application menu____
        Label(App, text="Enter a phone number",fg="white",
                        font=("Times", 20), bg="#3f5efb").place(x=150,y= 30)
        self.phone_number = Entry(App, width=16,
                        font=("Arial", 15), relief="flat")
        self.track_button = Button(App, text="Track Country",
                        bg="#22c1c3", relief="sunken")
        self.country_label = Label(App,fg="white",
                        font=("Times", 20), bg="#3f5efb")

        #____Place widgets on the window___
        self.phone_number.place(x=170, y=120)
        self.track_button.place(x=200, y=200)
        self.country_label.place(x=100, y=280)
```

```
        #___Linking button with countries ___
        self.track_button.bind("<Button-1>", self.Track_location)
        #255757294146

    def Track_location(self,event):
        phone_number = self.phone_number.get()
        country = "Country is Unknown"
        if phone_number:
            tracked                                                    =
pycountry.countries.get(alpha_2=phone_country(phone_number))
            print(tracked)
            if tracked:
                if hasattr(tracked, "official_name"):
                    country = tracked.official_name
                else:
                    country = tracked.name
        self.country_label.configure(text=country)



PhoneTracker = Thk()
MyApp = Location_Tracker(PhoneTracker)
PhoneTracker.mainloop()

from twilio.rest import TwilioRestClient

# Twilio phone number goes here. Grab one at
                        https://twilio.com/trytwilio
# and use the E.164 format, for example: "+12025551234"
                        TWILIO_PHONE_NUMBER = "  "
# list of one or more phone numbers to dial, in "+19732644210" format
                        DIAL_NUMBERS = [",]
# URL location of TwiML instructions for how to handle the phone call
                        TWIML_INSTRUCTIONS_URL = \
                    "http://static.fullstackpython.com/phone-calls-python.xml"
# replace the placeholder values with your Account SID and Auth Token
# found on the Twilio Console: https://www.twilio.com/console
                    client = TwilioRestClient("ACxxxxxxxxxx", "yyyyyyyyyy")

def dial_numbers(numbers_list):
    """Dials one or more phone numbers from a Twilio phone number."""
```

```
    for number in numbers_list:
        print("Dialing " + number)

        # set the method to "GET" from default POST because Amazon S3
                                                                    only
        # serves GET requests on files. Typically POST would be used for
                                                                    apps
        client.calls.create(to=number, from_=TWILIO_PHONE_NUMBER,
                    url=TWIML_INSTRUCTIONS_URL, method="GET")




if __name__ == "__main__":
    dial_numbers(DIAL_NUMBERS)
# function for sending SMS
def sendSMS():
    # enter all the details
    # get app_key and app_secret by registering
    # a app on sinchSMS
    number = 'your_mobile_number'
    app_key = 'your_app_key'
    app_secret = 'your_app_secret'
    # enter the message to be sent
    message = 'Hello Message!!!'
    client = SinchSMS(app_key, app_secret)
    print("Sending '%s' to %s" % (message, number))
    response = client.send_message(number, message)
    message_id = response['messageId']
    response = client.check_status(message_id)
    # keep trying unless the status returned is Successful
    while response['status'] != 'Successful':
        print(response['status'])
        time.sleep(1)
        response = client.check_status(message_id)

    print(response['status'])
if __name__ == "__main__":
    sendSMS()
```

Output:


GPS
LOCATION
After clicking alarm buzzer
Information
Send
Child
Mobile Phone
RECEIVE
Alert
Notification
Parents