15L513–INTERNSHIP-II & INNOVATION PRACTICES

FACULTY ALERT SYSTEM

Samuel Priyadarshan S 17L242
Saran Raj G 17L243

Project report submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

of Anna University

OCTOBER 2019

PSG COLLEGE OF TECHNOLOGY
(Autonomous Institution)
COIMBATORE – 641 004

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

FACULTY ALERT SYSTEM

Bona fide record of work done by

Samuel Priyadarshan S 17L242
Saran Raj G 17L243

Project report submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

of Anna University

OCTOBER 2019

...………………………	……………………….	……..……………….
Mrs. P. Prabavathi	**Dr. D. Sivaraj**	**Dr. S. SUBHA RANI**
Faculty guide	Faculty guide	Head of the Department

Certified that the candidate was examined in the viva-voce examination held on ………………….

…………………………..
(External Examiner)

CONTENTS

CHAPTER **Page**
No.

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to our beloved Principal Dr. K. Prakasan for providing an opportunity and necessary facilities in carrying out this project work.

We also wish to express our sincere thanks to Dr. S. Subha Rani, Head, Department of Electronics and Communication Engineering, for the encouragement and support that she extended towards this project work.

Our sincere thanks to Dr. D. Sivaraj, Assistant Professor, Department of ECE and Mrs. P. Prabavathi, Assistant Professor, Department of ECE whose guidance and continuous encouragement throughout the course made it possible to complete this project work well in advance.

We are grateful to the technical staffs who also played a vital role in helping us with providing lab facilities, without which we would not have completed this project work.

Finally, I would like to thank all my student colleagues who have helped us in completing this project without any hassles.

ABSTRACT

In an educational institution, it can be quite necessary for a student to meet a faculty for various purposes. But the faculty on the other hand will have a busy schedule and it can be quite difficult to meet them at times of emergencies. Sometimes, meeting a faculty in the cabin can be piece of cake. The student does to the cabin, the faculty is available and the student gets the job done. But most of the times, it doesn't happen that way. Sometimes students have to wait for 2 days to meet the concerned faculty just for a 2 minute work.

To avoid this time wastage, we have come up with the idea of faculty alert system. Using this system, the student can be acknowledged if the faculty is in the cabin or not. This system has been implemented using NFC Technology and a GSM module. With this system a lot of time can be saved and it will be highly beneficial for students who meet the faculty often.

Let us consider a scenario where a student goes to meet a concerned faculty. If the concerned faculty is not available, the student has to feed his/her number through a PC. The student can now go and attend the class. When the faculty arrives, he/she opens the door with an NFC tag, using an NFC reader. The door now opens and hence this sends an alert message to the students who have proposed to meet the faculty using a GSM Module.

CHAPTER 1

INTRODUCTION

1.1 NFC TECHNOLOGY:

- NFC stands for Near Field Communication.
- Short-range radio technology that enables communication between devices that either touch or are momentarily held close together.
- Operates at 13.56 MHz frequency
- Data transmission rates such as 106 Kbps, 212 Kbps, and 424 Kbps.

1.2 OPERATING MODES:

1.2.1 Read/Write:

Read or write data to any of the supported tag types in a standard NFC data format.

1.2.2 Peer to Peer:

Two NFC-enabled devices can exchange data.

1.2.3 Card emulation:

An NFC-enabled device acts as reader when in contact with tags. The phone can act as a tag or contactless card for existing readers

1.3 TAGS AND READERS:

1.3.1 TAG:

The tag is a thin simple device containing antenna and small amount of memory. It is a passive device, powered by magnetic field. Depending on the tag type the memory can be read only, rewritable, and writable once

1.3.2 READER:

The reader is an active device, which generates radio signals to communicate with the tags. The reader powers the passive device in case of passive mode of communication.

1.4 WORKING OF NFC:

The active device like smartphone or tablet reads or writes the data on NFC tag using the principle of Electromagnetic Induction. Now, to read the information from the NFC tag, the active device generates the radio wave at the carrier frequency of 13.56MHz. These waves used to get coupled to the antenna of the passive tag and from this electromagnetic waves, NFC tag receives the power. This time-varying electromagnetic field generates the EMF or voltage in this passive tag. This generated voltage is rectified using and it is used to power up the chip inside the NFC tag. This small chip generally comprises of the clock, EEPROM, rectifier circuit and the small controller. Once the chip is powered up then the tag responds back to the active device by the technique which is known as the load modulation. In this technique, the tag uses the internal clock of the chip and it generates the auxiliary carrier frequency of 848KHz. Depending upon the stored data inside this chip, the load which is connected in parallel to the antenna is turned ON and OFF. And in this way, the data is sent back to the active device. In this way, by the principle of electromagnetic induction, the passive tag receives the power and using the load modulation the data is sent back to the active device.

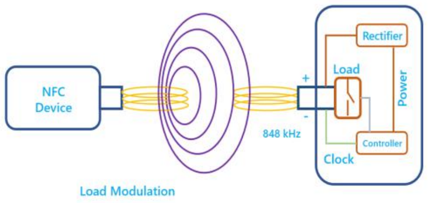

Fig. 1

CHAPTER 2

DESIGN AND METHODOLOGY

2.1 BLOCK DIAGRAM

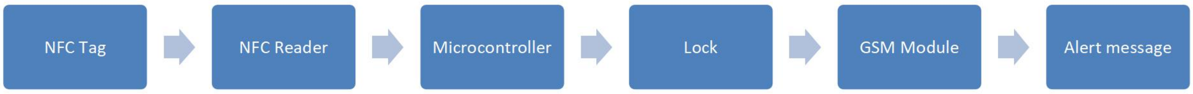

Fig 2.1

2.2 DESIGN:

We have designed our faculty alert system with the help of

- NFC Module (PN532)
- GSM Module (SIM800a)
- Arduino Uno R3

2.2.1 NFC Module (PN532):

NFC is a protocol designed for smart phones and similar devices to establish radio communication with each other by touching them together or bringing them into close proximity, usually no more than a few centimeters. This type of communication is needed for simple and fast data exchange between devices. And for engineers and hobbyists it is another important communication method. And for establishing this NFC communication we can use this **PN532 MODULE**. This module acts a modem to send and receive data. PN532 is NFC RFID

Fig. 2.2

(Radio Frequency Identification) module. This module is built around NXP **PN532 IC**. NXP PN532 IC is very popular in NFC applications. In our project, we have used it to get the data from the NFC tag, which is fed into the microcontroller.

2.2.2 GSM Module (SIM800a):

A GSM module or a GPRS module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system. The modem (modulator-demodulator) is a critical part here. These modules consist of a GSM module or GPRS modem powered by a power supply circuit and communication interfaces (like RS-232, USB 2.0, and others) for computer. A GSM modem can be a dedicated modem device with a serial,USB or Bluetooth connection, or it can be a mobile phone that provides GSM modem capabilities. A GSM module or GPRS modules are similar to modems, but there’s one difference: A GSM/GPRS Modem is an external equipment, whereas the GSM/GPRS module is a module that can be integrated within an equipment. It is an embedded piece of hardware. A GSM mobile, on the other hand, is a complete system in itself with embedded processors that are dedicated to provide an interface between the user and the mobile network. In our project, we have used this GSM Module to send the alert message to the student who has requested to meet the faculty. The mesage is sent as soon as the faculty scans the tag in the NFC Reader.

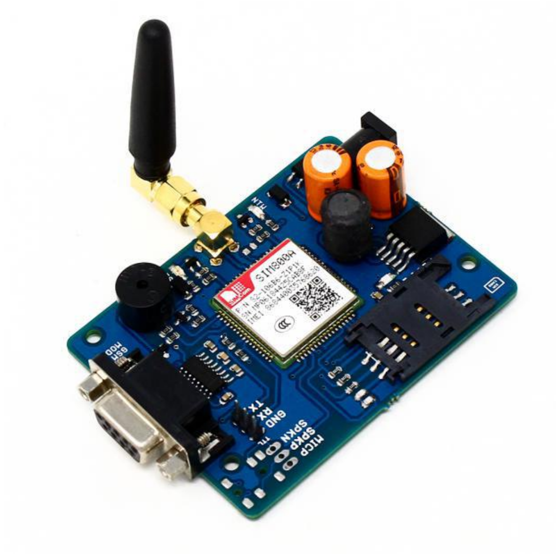

Fig 2.3

2.2.3 ARDUINO UNO R3:

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. In our project, we have used it as a microcontroller, which can get the data from the NFC reader and make sure that the alert message is sent to the concerned person, if requested.

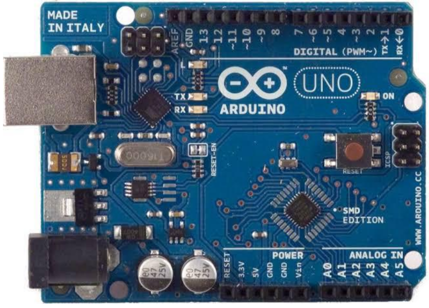

Fig 2.4

2.3 FLOW CHART:

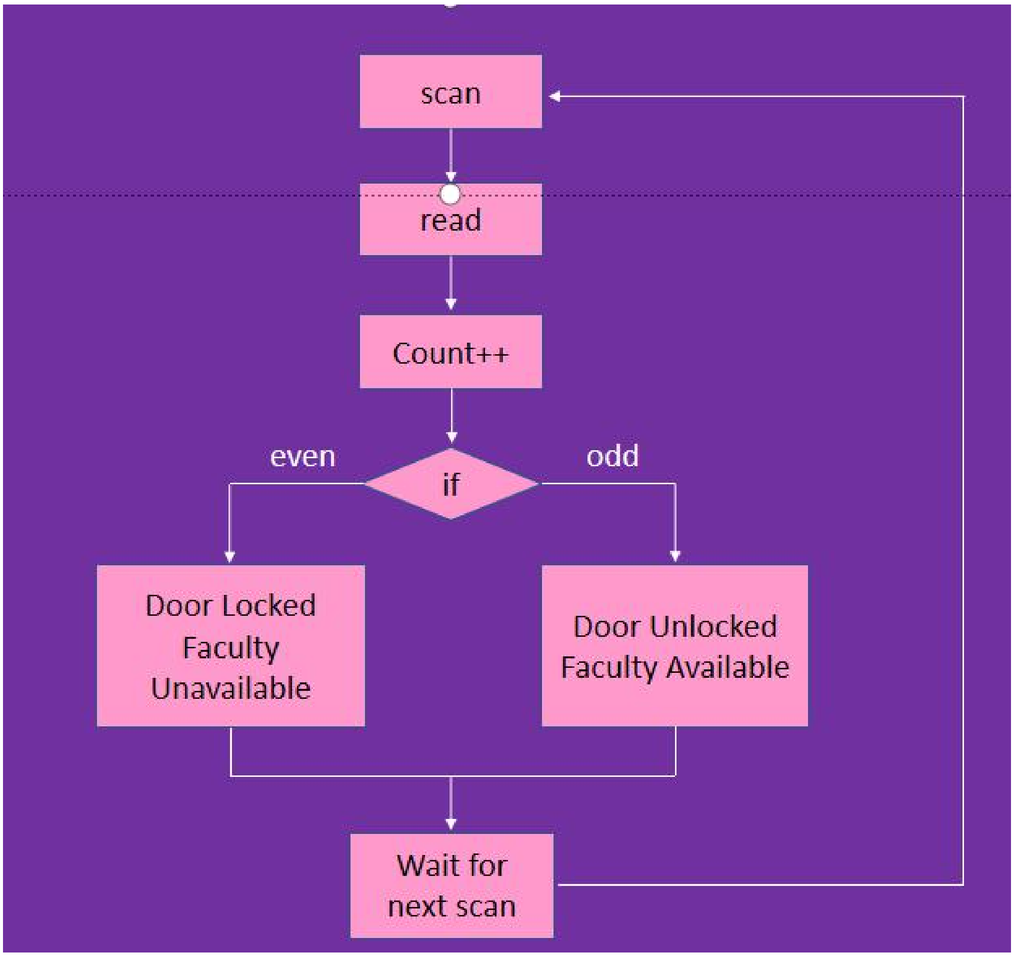
scan
read
Count++
even
if
odd
Door Locked Faculty Unavailable
Door Unlocked Faculty Available
Wait for next scan

Fig 2.5

CHAPTER 3

IMPLEMENTATION

3.1 CIRCUIT DIAGRAM:

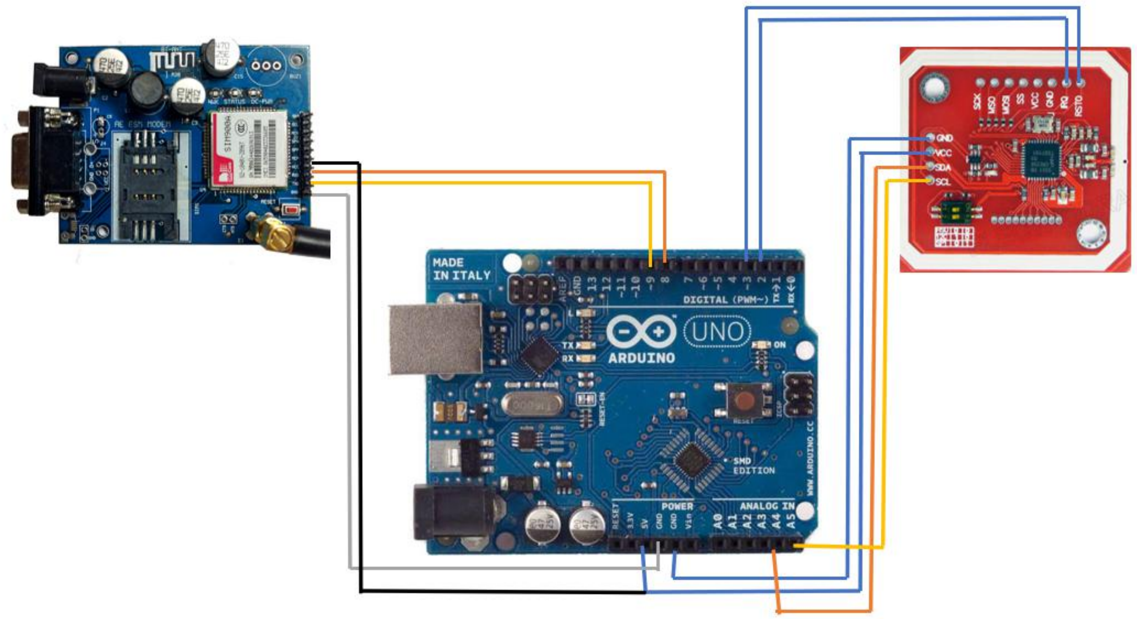

Fig 3

CHAPTER 4

RESULT

The proposed system is implemented and we have developed a proof of concept.

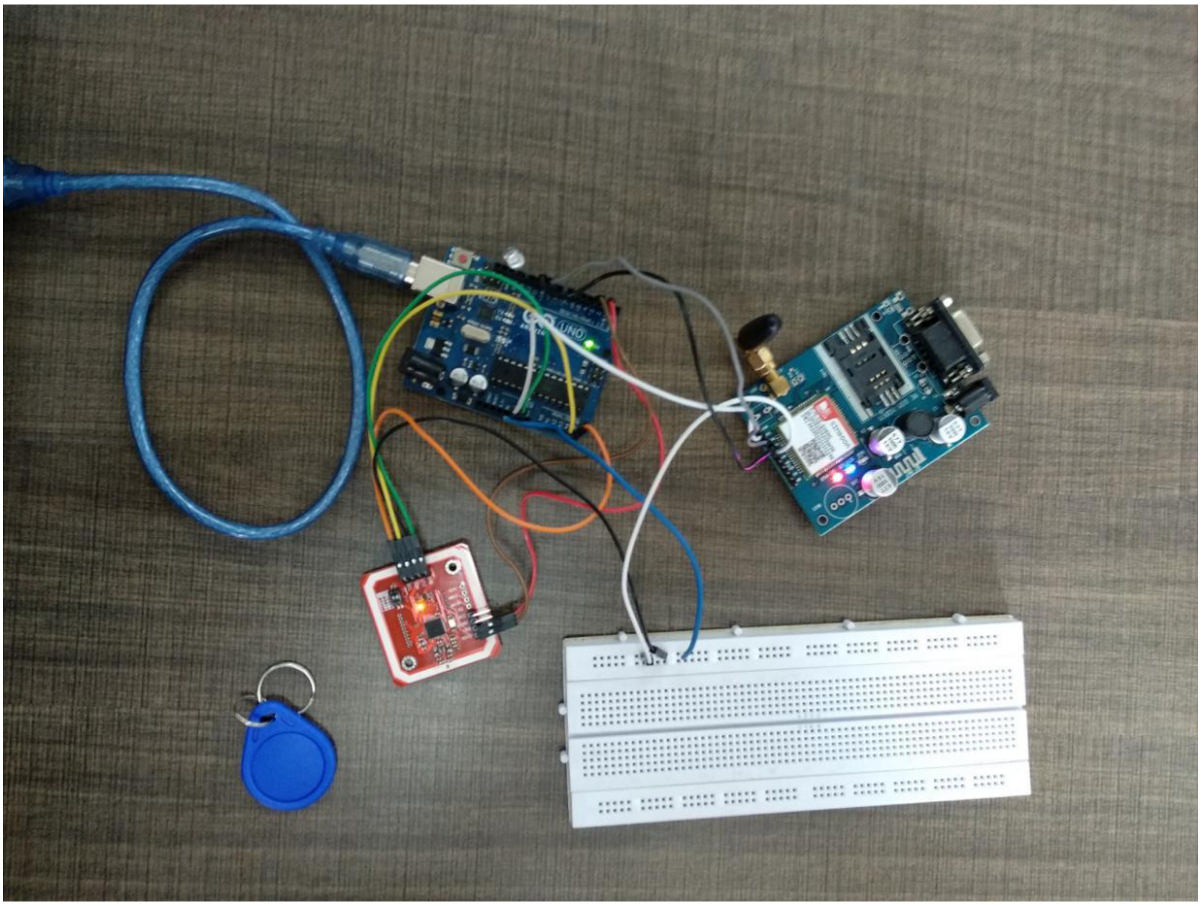

Fig 4

CHAPTER 6

CONCLUSION AND FUTURE ASPECT

This model will be useful for all the persons or even other faculties who are in need of meeting a concerned faculty. The waiting time to meet a faculty can be decreased exponentially.

Further, in the near future we are planning to bring a smart appointment system which can reduce the waiting time to a small extent, in case of multiple requests to meet a concerned faculty.

APPENDIX

CODE:

```
String temp,format1,format2;
String number1="+91";
String number2="+91";
char c;
int count=0;
int digit = 0;
int i,flag1,flag2,ok;

#include <Wire.h>
#include <PN532_I2C.h>
#include <PN532.h>   // The following files are included in the libraries Installed
#include <NfcAdapter.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(9, 10);


PN532_I2C pn532_i2c(Wire);
NfcAdapter nfc = NfcAdapter(pn532_i2c);  // Indicates the Shield you are using
//int count=0;
int LED = 13;

void setup(void) {
  pinMode(LED, OUTPUT);
  Serial.begin(115200);
  Serial.println("NFC TAG READER");
  mySerial.begin(115200);
  delay(500);// Header used when using the serial monitor
  nfc.begin();
}

void loop(void) {
  nfc_reader();
  Serial.print("press n to place your request");
  delay(2000);
  switch(Serial.read()){
    case 'n':
      ph_no1();
      while(ok){
          setnumber1();
          break;
   }
  }
 while(flag1){

  nfc_reader();
  Serial.print("press n to place your request");
  delay(2000);
  switch(Serial.read()){
    case 'n':
      ph_no2();
      while(ok){
          setnumber2();
          break;
   }
  }

  while (flag2){
    nfc_reader();
    }
  }

  number1 = "+91";
  number2 = "+91";
}

void nfc_reader(){
  Serial.println("\nScan your NFC tag on the NFC Shield\n");  // Command so that you an others will know what to do

  if (nfc.tagPresent())
  {
    NfcTag tag = nfc.read();

    if (tag.hasNdefMessage()) // If your tag has a message
    {

      NdefMessage message = tag.getNdefMessage();
      Serial.print("\nThis Message in this Tag is ");
      Serial.print(message.getRecordCount());
      Serial.print(" NFC Tag Record");
      if (message.getRecordCount() != 1) {
        Serial.print("s");
      }
      Serial.println(".");

      // If you have more than 1 Message then it wil cycle through them
      int recordCount = message.getRecordCount();
      int i=0;

      {
        NdefRecord record = message.getRecord(i);

        int payloadLength = record.getPayloadLength();
        byte payload[payloadLength];
        record.getPayload(payload);

        Serial.println(payloadLength);



        String payloadAsString = ""; // Processes the message as a string vs as a HEX value
        for (int c = 0; c < payloadLength; c++) {

          if (c<3){
            continue;
            }
           else{
          payloadAsString += (char)payload[c];
           }
        }
        Serial.print("  Information (as String): ");
        Serial.println(payloadAsString);

if (payloadAsString == "17L242"){
  count+=1;
  if (count==1){
    Door_open();
    GSMEntry1();
    GSMEntry2();
    }
   if (count==2){
    Door_close();
    GSMExit1();
    GSMExit2();
    count=0;
    flag1 = 0;
    flag2 = 0;
    }

  }

      }
    }
  }
  delay(1000);
  //while(1);
  }

void GSMEntry1() {
  Serial.println("Success");

      mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
     delay(1000);  // Delay of 1 second
     mySerial.println(format1); // Replace x with mobile number
     delay(1000);
     mySerial.println("Faculty is in cabin.");// The SMS text you want to send
     delay(100);
     mySerial.println((char)26);// ASCII code of CTRL+Z for saying the end of sms to  the module
     delay(1000);
     //while (1);
}

void GSMEntry2() {
  Serial.println("Success");
```

```
      mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
     delay(1000);  // Delay of 1 second
     mySerial.println(format2); // Replace x with mobile number
     delay(1000);
     mySerial.println("Faculty is in cabin.");// The SMS text you want to send
     delay(100);
     mySerial.println((char)26);// ASCII code of CTRL+Z for saying the end of sms to  the module
     delay(1000);
     //while (1);
}
void Door_open(){

    Serial.println("Door open");
    digitalWrite(LED, HIGH);
    delay(1000);
    delay(100);
}

void Door_close(){

  Serial.println("Door close");
  digitalWrite(LED, LOW);
  delay(100);
  digitalWrite(LED, HIGH);
  delay(100);
  digitalWrite(LED, LOW);
  delay(100);
  digitalWrite(LED, HIGH);
  delay(100);
  digitalWrite(LED, LOW);
    delay(100);
```

```
}
void GSMExit1() {
  // put your main code here, to run repeatedly:
      mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
     delay(1000);  // Delay of 1 second
     mySerial.println(format1); // Replace x with mobile number
     Serial.println(format1);
     delay(1000);
     mySerial.println("Sorry, The faculty left.");// The SMS text you want to send
     delay(100);
     mySerial.println((char)26);// ASCII code of CTRL+Z for saying the end of sms to  the module
     delay(1000);

}

void GSMExit2() {
  // put your main code here, to run repeatedly:
      mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
     delay(1000);  // Delay of 1 second
     mySerial.println(format2); // Replace x with mobile number
     Serial.println(format2);
     delay(1000);
     mySerial.println("Sorry, The faculty left.");// The SMS text you want to send
     delay(100);
     mySerial.println((char)26);// ASCII code of CTRL+Z for saying the end of sms to  the module
     delay(1000);

}

String ph_no1()
{
   Serial.println("Enter number");
  delay (10000);
  for(i=0; i<20; i++){
    c = (char)Serial.read();
    delay(300);
    if (isDigit(c))
     {
       number1 += c;
       //digit += 1;
     }

  }
  Serial.println(number1);
 delay(5000);
 ok=1;
 return number1;
  }

String setnumber1(){

 format1= "AT+CMGS="   ;


  format1 = format1 + '"' + number1 + '"' ;
   flag1 = 1;
  Serial.println(format1);
  delay(1000);
  Serial.println("Your request has been placed");
  return format1;
  ok=0;
  }

 String ph_no2()
{
   Serial.println("Enter number");
   delay (10000);
   for(i=0; i<20; i++){
      c = (char)Serial.read();
      delay(300);
      if (isDigit(c))
        {
          number2 += c;
          //digit += 1;
        }

   }
   Serial.println(number2);
 delay(5000);
 ok=1;
 return number2;
  }

String setnumber2(){

 format2= "AT+CMGS="   ;

  format2 = format2 + '"' + number2 + '"' ;
   flag2 = 1;
  Serial.println(format2);
  delay(1000);
  Serial.println("Your request has been placed");
  return format2;
  ok=0;
  }
```

BIBLIOGRAPHY

- https://www.google.com/search?q=pn532+nfc+module&client=ms-unknown&biw=412&bih=652&tbm=isch&prmd=svin&sxsrf=ACYBGNRwOwNPGxU2FcAaDydMmEyb-ptxNg:1571737342891&source=lnms&sa=X&ved=0ahUKEwicmoHwya_lAhWTmuYKHbbgCfkQ_AUIEygD#imgrc=MQ2exlMo7pATcM
- https://www.proto-pic.co.uk/arduino-uno-r3.html
- https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcT_v1wGaV-ANjGpzS-PJP5X8vxmFjLFAcd9JQGwfNRCxU41r9yh
- https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcRZc4UUrhd0pLWAB2jlKdsmDaAF6TLiryH5SV1kHC2z5UnYD-bn
- https://components101.com/wireless/pn532-nfc-rfid-module
- https://robu.in/product/sim800a-quad-band-gsmgprs-module-rs232-interface/