

Fingerprint Based Student Attendance System

Abstract: The system includes terminal fingerprint acquisition module and attendance module. It can realize automatically such functions as information acquisition of fingerprint, processing, and wireless transmission, fingerprint matching and making an attendance report. After taking the attendance, this system sends the attendance of every student to their own mobile and their parent's mobile through API system and also stored the attendance of respective student to calculate the percentage of attendance and alerts to class in charge. Attendance system facilitates access to the attendance of a particular student in a particular class. This system eliminates the need for stationary materials and personnel for the keeping of records and efforts of class in charge.

Equipment:

- Optical Fingerprint Sensor Module-FPM10A\
- Arduino Uno\
- Buck converter\
- HC05 Bluetooth Module\
- TTL\
- Power supply(5V, 5A)\
- Relay
- Rechargeable Li-Po battery
- LED
- Connectors

Designing System Hardware:

Optical Fingerprint Sensor Module-FPM10A: For securing project with biometrics - this all-in-one optical fingerprint sensor will make adding fingerprint detection and verification super simple. These modules are typically used in safes - there's a high powered DSP chip that does the image rendering, calculation, feature-finding and searching. Connect to any microcontroller or system with TTL serial, and send packets of data to take photos, detect prints, hash and search .We can also enroll new fingers directly up to 162 finger prints can be stored in the on board FLASH memory. There's a red LED in the lens that lights up during a photo so we know its working .We like this particular sensor because not only is easy to use, it also comes with fairly straight-forward Windows software that makes testing the module simple .We can even enroll using the software and see an image of the fingerprint on our computer screen.

Features:

Supply voltage: 3.6 - 6.0VDC

Operating current: 120mA max

Peak current: 150mA max

Fingerprint imaging time: <1.0 seconds

Window area: 14mm x 18mm

Signature file: 256 bytes

Template file: 512 bytes

Storage capacity: 162 templates

Safety ratings (1-5 low to high safety)

False Acceptance Rate: <0.001% (Security level 3)

False Reject Rate: <1.0% (Security level 3)

Interface: TTL Serial

Baud rate: 9600, 19200, 28800, 38400, 57600 (default is 57600)

Working temperature rating: -20C to +50C

Working humidity: 40%-85% RH

Full Dimensions: 56 x 20 x 21.5mm

Exposed Dimensions (when placed in box): 21mm x 21mm x 21mm triangular

Weight: 20 grams

Source code:

Fingerprint sensor: Firstly we have to upload this code on arduino uno .The code is given below:

```
#include <Adafruit_Fingerprint.h>

// On Leonardo/Micro or others with hardware serial, use those! #0 is green wire, #1 is white
// uncomment this line:
// #define mySerial Serial1

// For UNO and others without hardware serial, we must use software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
// comment these two lines if using hardware serial
SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit finger detect test");

  // set the data rate for the sensor serial port
  finger.begin(57600);
```

```

if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
} else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
}

finger.getTemplateCount();
Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");
Serial.println("Waiting for valid finger...");
}

void loop()          // run over and over again
{
    getFingerprintIDez();
    delay(50);        //don't need to run this at full speed.
}

uint8_t getFingerprintID() {
    uint8_t p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println("No finger detected");
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Communication error");
            return p;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Imaging error");
            return p;
        default:
            Serial.println("Unknown error");
            return p;
    }

    // OK success!

    p = finger.image2Tz();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image converted");
            break;
        case FINGERPRINT_IMAGEMESS:
            Serial.println("Image too messy");
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Communication error");

```

```

    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);

return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return -1;

    // found a match!
    Serial.print("Found ID #"); Serial.print(finger.fingerID);
    Serial.print(" with confidence of "); Serial.println(finger.confidence);
    return finger.fingerID;
}

```

Enrolling: First is we'll need to enroll fingerprints that means assigning ID #'s to each print so we can query them later. Once we've enrolled all our prints, we can easily 'search' the sensor, asking it to identify which ID (if any) is currently being photographed .The code is:

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

uint8_t getFingerprintEnroll(int id);

// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  Serial.println("fingertest");

  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1);
  }
}

void loop()
{
  Serial.println("Type in the ID # you want to save this finger as...");
  int id = 0;
  while (true) {
    while (! Serial.available());
    char c = Serial.read();
    if (! isdigit(c)) break;
    id *= 10;
    id += c - '0';
  }
  Serial.print("Enrolling ID #");
  Serial.println(id);

  while (! getFingerprintEnroll(id) );
}
```

```

uint8_t getFingerprintEnroll(int id) {
    int p = -1;
    Serial.println("Waiting for valid finger to enroll");
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println(".");
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Communication error");
                break;
            case FINGERPRINT_IMAGEFAIL:
                Serial.println("Imaging error");
                break;
            default:
                Serial.println("Unknown error");
                break;
        }
    }
}

```

```

p = finger.image2Tz(1);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

```

```

Serial.println("Remove finger");
delay(2000);
p = 0;

```

```
while (p != FINGERPRINT_NOFINGER) {  
    p = finger.getImage();  
}
```

```
p = -1;  
Serial.println("Place same finger again");  
while (p != FINGERPRINT_OK) {  
    p = finger.getImage();  
    switch (p) {  
        case FINGERPRINT_OK:  
            Serial.println("Image taken");  
            break;  
        case FINGERPRINT_NOFINGER:  
            Serial.print(".");  
            break;  
        case FINGERPRINT_PACKETRECEIVEERR:  
            Serial.println("Communication error");  
            break;  
        case FINGERPRINT_IMAGEFAIL:  
            Serial.println("Imaging error");  
            break;  
        default:  
            Serial.println("Unknown error");  
            break;  
    }  
}
```

```
p = finger.image2Tz(2);  
switch (p) {  
    case FINGERPRINT_OK:  
        Serial.println("Image converted");  
        break;  
    case FINGERPRINT_IMAGEMESS:  
        Serial.println("Image too messy");  
        return p;  
    case FINGERPRINT_PACKETRECEIVEERR:  
        Serial.println("Communication error");  
        return p;  
    case FINGERPRINT_FEATUREFAIL:  
        Serial.println("Could not find fingerprint features");  
        return p;  
    case FINGERPRINT_INVALIDIMAGE:  
        Serial.println("Could not find fingerprint features");  
        return p;  
    default:  
        Serial.println("Unknown error");  
        return p;  
}
```

```
p = finger.createModel();
```

```

if (p == FINGERPRINT_OK) {
  Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
  Serial.println("Communication error");
  return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
  Serial.println("Fingerprints did not match");
  return p;
} else {
  Serial.println("Unknown error");
  return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
  Serial.println("Stored!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
  Serial.println("Communication error");
  return p;
} else if (p == FINGERPRINT_BADLOCATION) {
  Serial.println("Could not store in that location");
  return p;
} else if (p == FINGERPRINT_FLASHERR) {
  Serial.println("Error writing to flash");
  return p;
} else {
  Serial.println("Unknown error");
  return p;
}
}

```

Output:

COM8 (Arduino/Genuino Uno)

```

Found fingerprint sensor!
Found ID #1 with confidence of 96
Found ID #2 with confidence of 154
Found ID #3 with confidence of 83

```


Finger print module to Bluetooth connection: After the enrollment of fingerprints we have paired two Bluetooth module (HC-05) using one USB to TTL converter with the receiver module. For receiving the data the source code is given below:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
int fingerprintID = 0;
String IDname;
#include <SoftwareSerial.h>
SoftwareSerial BTserial(10,11); // RX | TX
// Connect the HC-05 TX to Arduino pin 10 RX.
// Connect the HC-05 RX to Arduino pin 11 TX through a voltage divider.
// MASTER

int c ;

void setup() {

  Serial.begin(56000);
  Serial.println("CLEARDATA");
  Serial.println("F ID:");
  Serial.println("LABEL,Time,Started Time,Register value");
  Serial.println("RESETTIMER");
  // Serial.begin(57600);

  Serial.println("Arduino is ready");
  Serial.println("Remember to select Both NL & CR in the serial monitor");

  // HC-05 default serial speed for AT mode is 38400
  BTserial.begin(57600);

  // Serial.begin(9600);
  // set the data rate for the sensor serial port
  finger.begin(56000);
```

```

if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
}
else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
}

```

```

//OLED display setup
Wire.begin();
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
//displays main screen
displayMainScreen();

}

```

```

void loop() {
    displayMainScreen();
    fingerprintID = getFingerprintIDez();
    delay(50);
    if(fingerprintID == 3){
        IDname = "Sara";
        displayUserGreeting(IDname);
    }
    if( fingerprintID == 2 ){
        IDname = "Ali";
        displayUserGreeting1(IDname);
    }
    else if(fingerprintID == 4){
        IDname = "Khan";
        displayUserGreeting2(IDname);
    }
}
}

```

```

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return -1;
}

```

```

// found a match!
Serial.print("Found ID #");
Serial.print(finger.fingerID);
Serial.print(" with confidence of ");
Serial.println(finger.confidence);
return finger.fingerID;
}
void displayMainScreen(){

    delay(2000);

}

void displayUserGreeting(String Name){
    delay(1000);
    fingerprintID = 0;
    int sensorValue=finger.fingerID;

    if (Serial.available())
    {
        sensorValue = Serial.read();
        BTserial.write(sensorValue);
    }

    Serial.print("DATA,TIME,TIMER,");
    Serial.print("F ID:");
    Serial.println(sensorValue );

}

////////////////////////////////////
void displayUserGreeting1(String Name){

    delay(1000);
    fingerprintID = 0;
    int sensorValue=finger.fingerID;

    if (Serial.available())
    {
        sensorValue = Serial.read();
        BTserial.write(sensorValue);
    }

    Serial.print("DATA,TIME,TIMER,");
    Serial.print("F ID:");
    Serial.println(sensorValue );

```

```

}
void displayUserGreeting2(String Name){

    delay(1000);
    fingerprintID = 0;
    int sensorValue=finger.fingerID;

    if (Serial.available())
    {
        sensorValue = Serial.read();
        BTserial.write(sensorValue);
        Serial.print("DATA,TIME,TIMER,");
        Serial.print("F ID:");
        Serial.println(sensorValue );
    }

}

```

Transferring received data to MySQL Server: After receiving the data through TTL the data was stored in MySQL server. Our first task was to make a dialogue box using visual basic. The dialogue box is used to connect TTL port and show the received data from finger print module. The next task was to create a database that contains all the information of students. There are several tables and a view which is use to show the attendance reports of the students. When any student place his/her finger on finger print module then the matched ID will show in the dialogue box through TTL and the ID will also store in the database.

The screenshot displays the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left shows the database structure for 'OVZZZ.fingerprint'. The 'Query Designer' window in the center shows the 'dbo.Information' table with the following data:

ID	Roll	Enroll_ID	Enroll_Name	Blood_Group	Mobile_No	Parent's_No	E-mail
1	1610046	46	Nuzhat Navar	O(+ve)	8801922029625	8801781395866	novanavar24@...
2	1610050	50	Amit Kumar Paul	AB(+ve)	8801781395866	8801922029625	amiteceruet@...
3	1610051	51	Methila Afrin Meem	N&L	8801743291743	N&L	N&L
4	1610052	52	Ovi Sarkar	N&L	8801827602947	8801711370233	ovisarkarecelan...
*	N&L	N&L	N&L	N&L	N&L	N&L	N&L

The 'Properties' window on the right shows the 'Query Designer' tab with the following settings:

- Destination Table: No
- Distinct Values: No
- GROUP BY Extens: <None>
- Output All Colum: Yes
- Query Parameters: No parameters have b
- SQL Comment: No
- Top Specification: No

Fig: Information table

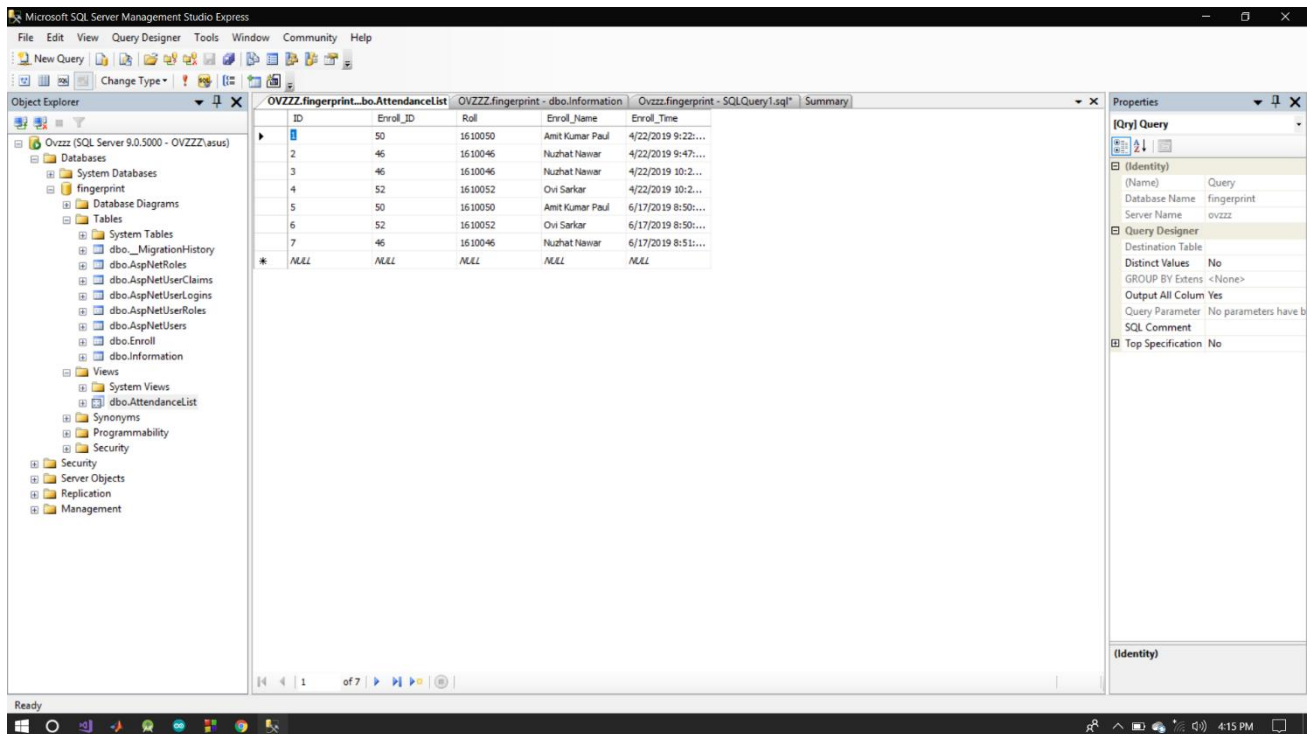


Fig: Attendance list (View)

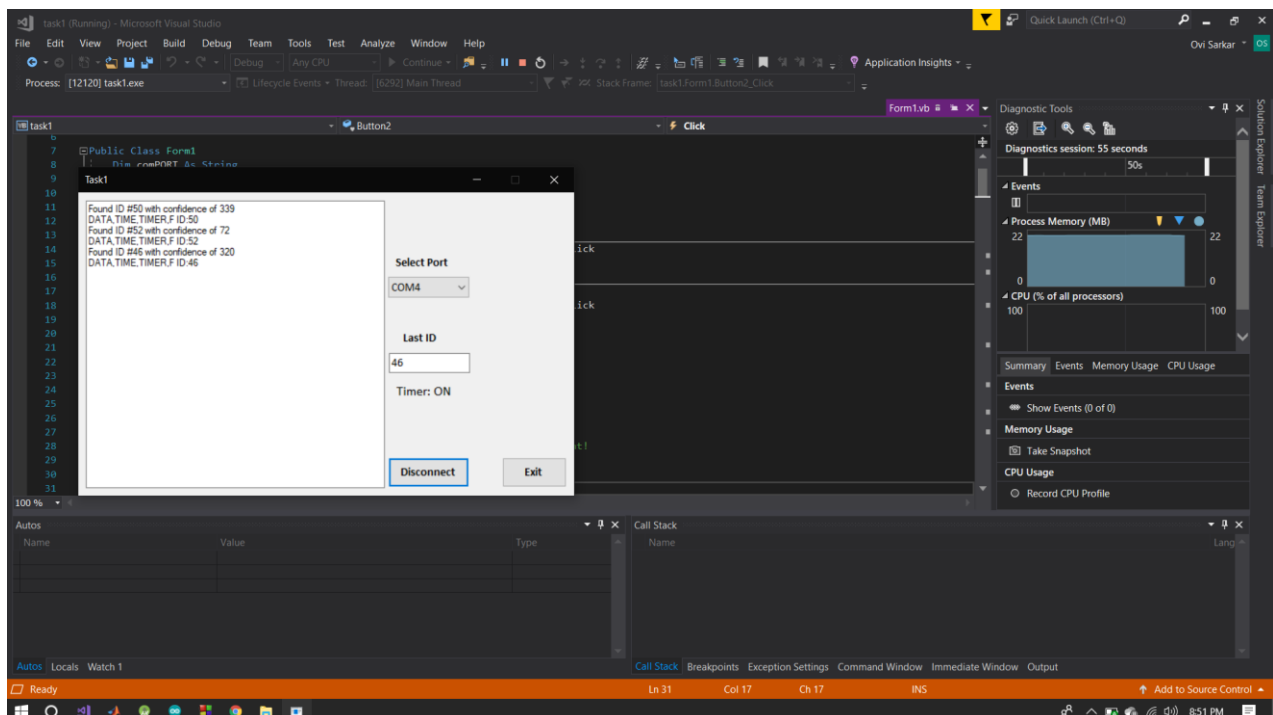


Fig: Dialogue box

The database is connected to server using C# through visual basic to show the stored data. A view is created which we can access through any internet browser. When the matched ID is stored in the database at the same time two messages are sent to corresponding two different contact numbers one is the student's own contact number and another one is guardian's contact number. The messages are sent via API system.

Index - Student Attendance System

localhost:5821/AttendanceLists

Home About Contact Attendance List

Hello demo@demo.com! Log off

Index

Enroll_ID	Roll	Enroll_Name	Enroll_Time
50	1610050	Amit Kumar Paul	4/22/2019 9:22:07 PM
46	1610046	Nuzhat Nawar	4/22/2019 9:47:21 PM
46	1610046	Nuzhat Nawar	4/22/2019 10:25:51 PM
52	1610052	Ovi Sarkar	4/22/2019 10:26:03 PM
50	1610050	Amit Kumar Paul	6/17/2019 8:50:36 PM
52	1610052	Ovi Sarkar	6/17/2019 8:50:47 PM
46	1610046	Nuzhat Nawar	6/17/2019 8:51:03 PM

Fig: Attendance list

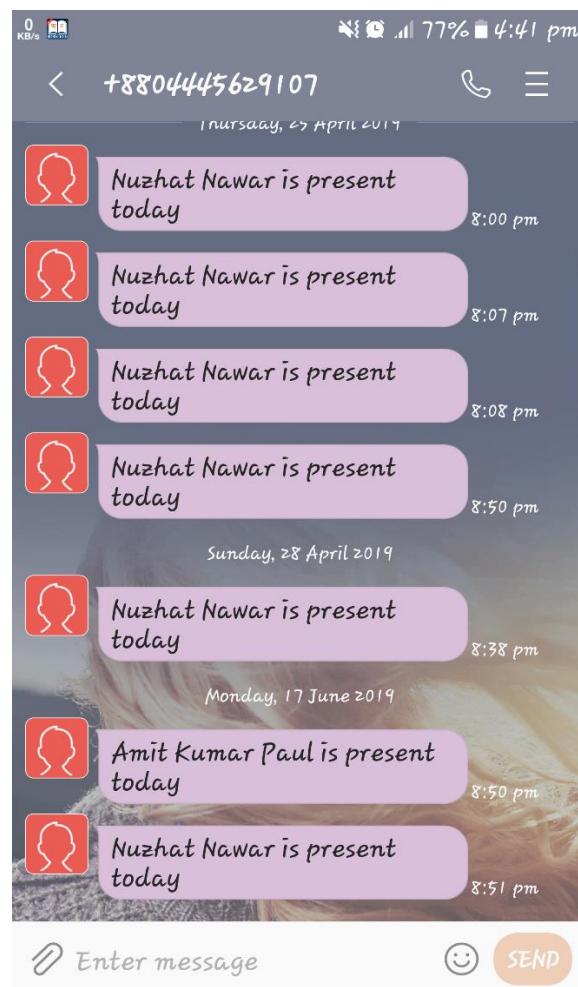


Fig: Message from one mobile

Discussions: The main purpose of this project is to monitor the student attendance in lecture, tutorial and laboratory sessions in more efficient way and send this attendance to their parents. This system resists students from bunking classes through SMS sending feature to parents. Biometrics has been used effectively for more than a decade for time and attendance system. Fingerprint attendance system is a cost effective simplified system that uses fingerprints for identification.