RFID BASED ATTENDANCE MANAGEMENT SYSTEM USING ARDUINO

Project Report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

In

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted By

SHUBHAM KUMAR MITRA	University Roll No. 12016009002188
SOURABH KUMAR RAVI	University Roll No. 12016009002094
ANURAG SINGH	University Roll No. 12016009002039
SAAMIR HAQUE MOLLA	University Roll No. 12016009002068
SWARNAK DEY	University Roll No. 12016009002019

Under the guidance of

PROF. KARUNAMAYEE DEY

Department of Electronics & Communication Engineering

UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160.

CERTIFICATE

This is to certify that the project titled **RFID BASED ATTENDANCE MANAGEMENT SYSTEM USING ARDUINO** submitted by **SHUBHAM KUMAR MITRA (University Roll No. 12016009002188), SOURABH KUMAR RAVI (University Roll No. 12016009002094)**, **ANURAG SINGH (University Roll No. 12016009002039), SAAMIR HAQUE MOLLA (University Roll No. 12016009002068)** and **SWARNAK DEY (University Roll No. 12016009002019).** Students of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, as submitted for the partial fulfilment of requirements for the degree of Bachelor of Technology in Electronics & Communication Engineering, is a bona fide work carried out by them under the supervision and guidance of Prof. KARUNAMAYEE DEY during 7th and 8th Semester of academic session of 2018-2019. The content of this report has not been submitted to any other university or institute for the award of any other degree.

I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

Prof. Karunamayee Dey
Assistant Professor
Department of ECE
UEM, Kolkata

Prof. Dr. Soumen Banerjee
Head of the Department
Department of ECE
UEM, Kolkata

ACKNOWLEDGEMENT

We would like to take this opportunity to thank and acknowledge with due courtesy everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide Prof. Karunamayee Dey of the Department of Electronics & Communication Engineering, UEM, Kolkata, for her wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to Prof. Dr. Soumen Banerjee, HOD, Electronics & Communication Engineering, UEM, Kolkata and all other departmental faculties for their ever-present assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

_____ _____

Shubham Kumar Mitra Sourabh Kumar Ravi

_____ _____

Anurag Singh Saamir Haque Molla

_____

Swarnak Dey

TABLE OF CONTENTS

ABSTRACT

In recent years, there has been a rise in the number of applications based on Radio Frequency Identification (RFID) systems and have been successfully applied to different areas as diverse as transportation, health-care agriculture and hospitality industry to name a few.

RFID technology is an automatic wireless identification system that works by the help of active and passive cards and a reader. In this work we have tried to ease the problem of manually taking attendance with the use of RFID technology. This system will help the authority manage the attendance system in a more organized, efficient and time saving manner. The proposed method has been implemented in a prototype system that has proved the effectiveness of the system in easing the chores of attendance taking as a result of the automation of the system using the RFID technology. The design of the system is simple, inexpensive and portable making it a good candidate for commercial and academic purposes.

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In this project we will be designing RFID Based Attendance System using Arduino.

EM-18 RFID Reader is a very simple yet effective module. It is an RFID module and is used for scanning RFID cards. It's a new technology and is expanding day by day. Now-a-days it is extensively used in offices where employees are issued an RFID card and their attendance is marked when they touch their card to RFID reader. We have seen it in many movies that when someone places one's card over some machine then the door opens or closes. In short, it's a new emerging technology which is quite useful.

Radio frequency identification uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID tag consists of a tiny radio transponder; a radio receiver and transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data, usually an identifying inventory number, back to the reader. This number can be used to inventory goods. There are two types. *Passive tags* are powered by energy from the RFID reader's interrogating radio waves. *Active tags* are powered by a battery and thus can be read at a greater range from the RFID reader; up to hundreds of meters. Unlike a barcode, the tag doesn't need to be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method of automatic identification and data capture (AIDC).

RFID tags are used in many industries. For example, an RFID tag attached to an automobile during production can be used to track its progress through the assembly line; RFID-tagged pharmaceuticals can be tracked through warehouses; and implanting RFID microchips in livestock and pets enables positive identification of animals.

Since RFID tags can be attached to cash, clothing, and possessions, or implanted in animals and people, the possibility of reading personally-linked information without consent has raised serious privacy concerns. These concerns resulted in standard specifications development addressing privacy and security issues. ISO/IEC 18000 and ISO/IEC 29167 use on-chip cryptography methods for untraceability, tag and reader authentication, and over-the-air privacy. ISO/IEC 20248 specifies a digital signature data structure for RFID and barcodes providing data, source and read method authenticity. This work is done within ISO/IEC JTC 1/SC 31 Automatic identification and data capture techniques. Tags can also be used in shops to expedite checkout, and to prevent theft by customers and employees.

In 2014, the world RFID market was worth US$8.89 billion, up from US$7.77 billion in 2013 and US$6.96 billion in 2012. This figure includes tags, readers, and

software/services for RFID cards, labels, fobs, and all other form factors. The market value is expected to rise from $12.08bn in 2020 to US$16.23 billion by 2029.

CHAPTER 2: LITERATURE SURVEY

RFID Based Attendance System Review of Related Literature - RFID stands for Radio-frequency identification; it uses electromagnetic fields to automatically identify and track tags attached to objects.

2.1 On Student Monitoring using RFID -

According to Kassim M. (2012), this paper describes the development of a student attendance system based on Radio Frequency Identification (RFID) technology. The existing conventional attendance system requires students manually sign the attendance sheet every time they attend a class. Having a system that can automatically capture student's attendance by flashing their student card at the RFID reader can really save all the mentioned troubles. This is the main motive of the system and in addition having an online system accessible anywhere and anytime can greatly help the lecturers to keep track of their students attendance looking at a bigger picture, deploying the system throughout the academic faculty will benefit the academic management as students attendance to classes is one of the key factor in improving the quality of teaching and monitoring their student's performance. Besides, this system provides valuable online facilities to related academic management staff especially for the purpose of student's progress monitoring.

2.2 School of Information Technology Faculty Monitoring using Radio Frequency Identification (RFID) -

According to Justin Lee and et.al (2013), this study was to improve the faculty monitoring of the University of Baguio in the School of Information Technology Department. This helps the Student Assistant to easily locate the Faculty Members if needed and can generate a summary report. This may also lessen the tardiness of Faculty Members.

RFID systems also provide good personal security access to confidential data. (ubsit-fms-rfid.weebly.com)

2.3 Development of Student Monitoring System with the use of Low Frequency Radio Frequency Identification (RFID) and Short Messaging Service (SMS) -

According to Rhowel M. Delosa (2011), this study aimed to develop a Student Monitoring System using low frequency Radio Frequency Identification (RFID) and Short Messaging Service (SMS) in order to keep track of the students within the school premises. A computer program is to be developed to interact with the system. A series of tests was conducted to prove the accuracy of the entire system. With the aid of Chi-square test, the researcher determined the significant difference between the observed and expected data. The developed software can capture and record the name, entry and exit time of students. Moreover, the developed software can monitor the entry and exit time, account balance and schedules of current classes of the student thru Short Messaging Service (SMS). Future studies related to the topic may focus on the use of high frequency radio frequency (RF) readers instead of low frequency radio frequency (RF) readers. This will facilitate convenience to the students by just wearing or hanging the identification card with an RF tag every time the student enters and exits the school premises. (ejournals.ph).

CHAPTER 3: SCOPE OF STUDY

3.1 RFID in production - Inside the stores of the production department the stock level can be quickly monitored which helps in a real-time and efficient inventory management. The location and movement of the components, semi-finished and the finished products can be detected by the RFID system, which helps in production monitoring and control. The performance of departments and the individuals can be evaluated by the rate of movement of the garment components from the departments or individuals. This helps in improving the productivity and the quality. In the packing floor the mixing of different styles and sizes can be avoided by RFID tags. Also, the number of pieces in packed cartons can be counted without opening, which saves time and labour.

3.2 RFID in retail - It can be used in retail to monitor and control the floor level out-of-stocks (OOS). It is mostly required where there is high product display density, low staffing level and chances of mishandling is very high. The stock level of the items is properly maintained and the items can be grouped according to their demand. Similarly, while receiving the goods at the store the accuracy of the deliveries can be ensured quickly with less labour. Mostly the retailers verify the accuracy and integrity of the shipment by open-box audit prior to stocking or storing of goods. But now it is possible to read all the items packed inside the carton without opening, which saves time and labour cost. So, the RFID provides a new horizon to the retailers in receiving materials, which eliminates the invoice disputes, mixing of different quality products, etc.

3.3 Other applications – RFID has also many other applications in various fields. Low-frequency RFID tags are commonly used for animal identification. Pets can be implanted with small chips so that they may be returned to their owners if lost. High-frequency RFID tags are used in library book tracking, object tracking in stores,

building access control, airline baggage tracking, etc. The American Express Blue credit card now includes a high-frequency RFID tag.

The RFID tags can be used for toll collection at toll booths. The tags are read remotely as vehicles pass through the booths, and tag information is used to debit the toll from a prepaid account. The system helps to speed traffic at toll plazas. The smart key concept adopted by Toyota Prius and some Lexus models use an RFID circuit which allows the car to sense the presence of the key within 3 feet of the sensor. The driver can open the doors and drive the car when the key is inside the pocket or the handbag. The tag can be implanted on soldiers at the time of war to trace the missing soldiers.

3.4 Advantages of RFID over Barcodes –

Barcode systems though used for product information, inventory control, etc have some drawbacks as compared to RFID. The amount of information stored in a barcode is very less as compared to RFID. RFID can store up to 1000 bytes of data. An RFID is specific to each item, whereas the barcode is not. Barcode needs human interaction for proper operation. It requires time-of-sight access to an optical scanner for the product related information. The barcode is to be replaced if the information it contains needs modification, but in RFID it can be modified at stages of the supply chain by the interaction between the microchip and the reader software. The barcode system is less accurate as compared to RFID.

3.5 Limitations of RFID -

Though RFID technology has already been applied effectively, it has certain technological barriers that still need to be overcome to optimize its application. These lacunas are high investment, lack of security and privacy, and some related to the technology of RFID.

a. **Cost:** Although there is a great potential of RFID in the local logistics sector, the major drawback is the cost of the RFID tag, which is higher as compared to barcode systems. So industrial leaders are concerned about the return on investment and net profit by investing the extra cost in the existing system. The cost depends on the volume of usage. The lowest cost tags available on the market are as low as Rs 8 each in volumes of 1 crore units or more. The average cost of a RFID tag is around Rs 18 to Rs 25 if the volume is not sufficiently large but the cost is only Rs 2.50 for a barcode.

b. **Security and Privacy:** The security and privacy of the RFID against unauthorized readers is in debate from the very beginning. There is a great challenge to consumer privacy. The consumers using the product with RFID tags can be traced easily. The RFID tag broadcasts the ID serial number or the Electronic Product Code (EPC) to the nearby reader. There is a very high chance of privacy violations.

c. **Technology:** As the RFID is based on the concept of Radio Frequency, it can be interfered with other radio transmissions, metals, liquids, etc. The degree of interference depends upon the frequency of the tag and the usage environment.

CHAPTER 4: RADIO FREQUENCY IDENTIFICATION

RFID stands for Radio Frequency Identification. RFID is a means of identifying a person or object using Radio Frequency Transmission. RFID is used to collect information automatically by radio frequency data communication between a mobile object and an RFID reader to identify and track them. They are most commonly referred to as reader and tag respectively. There are Tags and Reader in the system. A typical reader is a device that has one or more antennas that emit radio waves and receive signals back from the tag. To retrieve the data stored on an RFID tag, a reader is needed. RFID is used to read or write information on a tag and pass that information to a system for storage and processing. Generally, RFID systems consist of 2 parts: Interrogator and Transponders. Interrogator and Transponder are also known as RFID Reader and RFID Tag respectively.

CHAPTER 5: COMPONENTS REQUIRED

To design RFID Based Attendance System using Arduino, we require the following components:

1. Arduino Uno Board

2. MFRC522 RFID Reader

3. GSM SIM900

5. VISUAL STUDIO (for GUI)

6. XAMPP PHPMyAdmin (for database)

5.1 COMPONENT DESCRIPTION

5.1.1 ARDUINO - Arduino is an open-source electronics platform based on easy-to- use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output – activating a motor, turning on a LED, publishing something online .You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects of complex scientific instruments .A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable,3Dprinting, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually

adapt them to their particular needs. The software, too, is open-source, and it is growing

through the contributions of users worldwide.

Figure 1

5.1.2 MF RC522 RFID READER:

MF RC522 is a highly integrated read and write card chip applied to the 13.56MHz contactless communication. Launched by the NXP Company, as a new member of the 13.56MHz highly integrated reader card series, MF RC522 is much similar to the existing MF RC500 and MF RC530 when there are also great differences. It communicates with the host machine via the serial manner which needs less wiring.

Figure 2

5.1.2.1 MF RC522 FEATURES:

- low-voltage
- low-cost
- small-sized non-contact card chip
- a best choice for intelligent instruments and portable handheld devices.

5.1.2.2 MF RC522 SPECIFICATIONS:

- Operating frequency: 13.56MHz.
- Supply Voltage: 3.3V.
- Current: 13-26mA.
- Read Range: Approx 3cm with supplied card and fob.
- SPI Interface.
- Max Data Transfer Rate: 10Mbit / s.

5.1.2.3 MF RC522 WORKING:

The MF RC522 uses advanced modulation and demodulation concepts which are fully presented in all types of 13.56MHz passive contactless communication methods and protocols. In addition, it supports a rapid CRYPTO1 encryption algorithm to verify MIFARE products. MFRC522 also supports MIFARE series of high-speed non-contact communication, with a two-way data transmission rate of up to 424kbit/s. It communicates with the host machine via the serial manner which needs less wiring.

5.1.3 SIM900A GSM MODULE:

The SIM900 is a complete Quad-band GSM/GPRS solution in a SMT module which can be embedded in the customer applications. Featuring an industry-standard interface, the SIM900 delivers GSM/GPRS 850/900/1800/1900MHz performance for voice, SMS, Data, and Fax in a small form factor and with low power consumption. With a tiny configuration, SIM900 can fit almost all the space requirements in M2M application, especially for slim and compact demand of design.

Figure 3

5.1.3.1 SIM900A FEATURES:

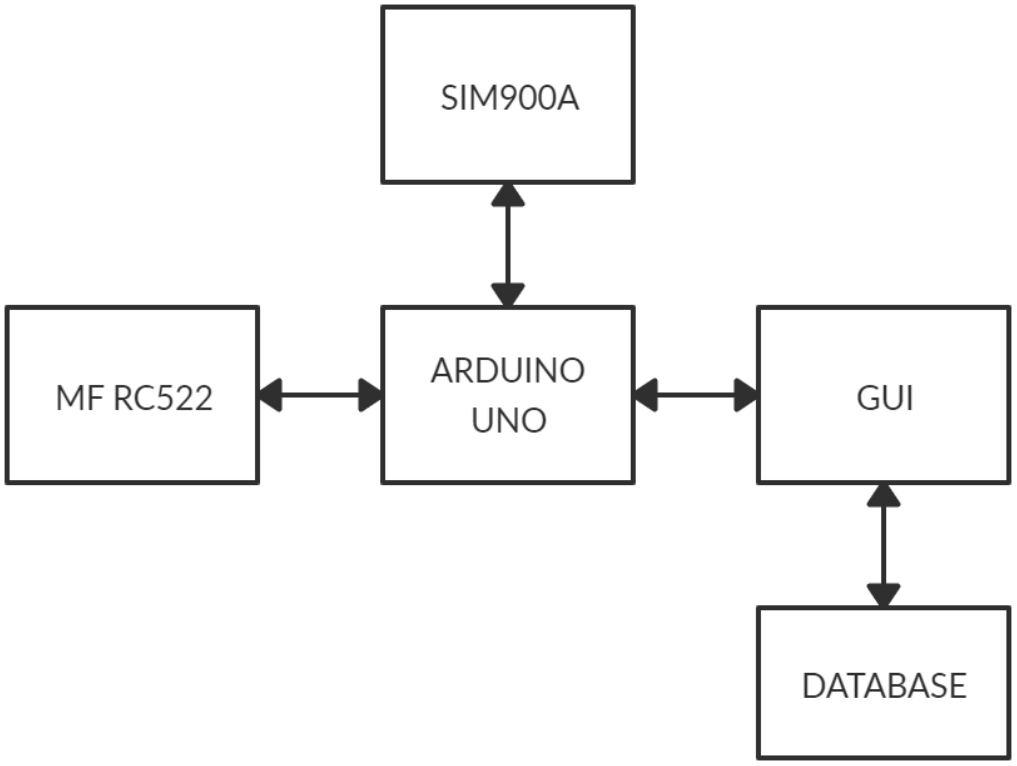
- SIM900 is designed with a very powerful single-chip processor integrating AMR926EJ-S core.
- Quad - band GSM/GPRS module with a size of 24mmx24mmx3mm.
- SMT type suit for customer application.
- An embedded Powerful TCP/IP protocol stack.
- Based upon a mature and field-proven platform, backed up by our support service, from definition to design and production.

5.1.3.2 SIM900A SPECIFICATIONS:

- Quad-Band 850/ 900/ 1800/ 1900 MHz
- GPRS multi-slot class 10/8
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
    - Class 4 (2 W @ 900 MHz)
    - Class 1 (1 W @ 1800MHz)
- Dimensions: 24x24x3mm
- Weight: 3.4g
- Control via AT commands (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands)
- SIM application toolkit
- Supply voltage range : 3.4V to 4.5V
- Low power consumption: 1.0mA(sleep mode)
- Operation temperature: -40°C to +85 °C
- Fax: Group 3, Class 1
- Data
    - GPRS class 10: max. 85.6 kbps (downlink)
    - PBCCH support
    - Coding schemes CS 1, 2, 3, 4
    - CSD up to 14.4 kbps
    - USSD
    - Non transparent mode
    - PPP-stack
    -

- SMS
  - Point to point MO and MT
  - SMS cell broadcast
  - Text and PDU mode
- Voice
  - Tricodec
    - Half rate (HR)
    - Full rate (FR)
    - Enhanced Full rate (EFR)
  - Hands-free operation (Echo suppression)
  - AMR
    - Half rate (HR)
    - Full rate (FR)

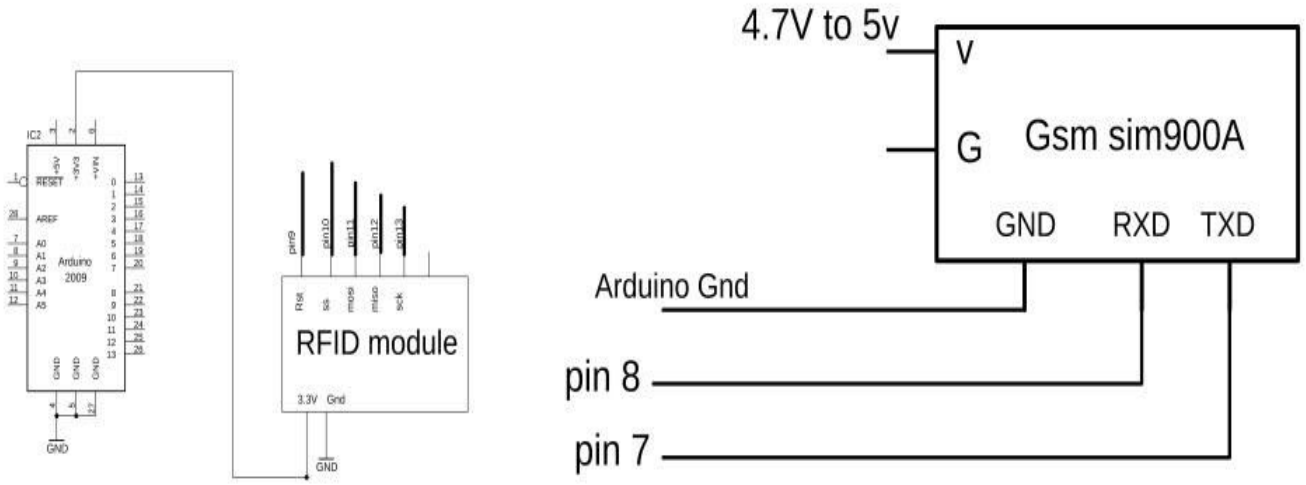
CHAPTER 6: BLOCK DIAGRAM –



This is the block diagram of RFID based attendance management system using Arduino, MF RC522 and SIM900A. Here Arduino UNO acts as a central processor for controlling all other components as an input/output unit.

We have used 5 Volt power supply to power all the components used in this project. RFID Reader module is interfaced with Arduino to read the data from RFID card/tag and send SMS to the corresponding phone number using SIM900A and also, store the attendance details of the student in the database which can be viewed from GUI.

CHAPTER 7: CIRCUIT DIAGRAM



CHAPTER 8: WORKING OF THE RFID SYSTEM

1. RFID (radio frequency identification) is a technique facilitating identification of any product or item without the requirement of any line of sight amid transponder and reader.

2. RFID Structure is continuously composed of 2 main hardware components. The transponder which is located on the product to be scanned and the reader which can be either just a reader or a read & write device, depending upon the system design, technology employed and the requirement. The RFID reader characteristically consists of a radio frequency module, a controlling unit for configurations, a monitor and an antenna to investigate the RFID tags. In addition, a number of RFID readers are in-built with an extra interface allowing them to forward the data received to another system (control system or PC).

3. RFID Tag – The actual data carrying tool of an RFID structure, in general comprises an antenna (coupling element) and an electronic microchip. In this project, we have used passive RFID tags.

CHAPTER 9: WORKING OF RFID BASED ATTENDANCE SYSTEM USING ARDUINO, MFRC522 AND SIM900A

In this project, we have designed an RFID based attendance system using Arduino as the microcontroller. First, we store a set of RFID card data in the Arduino. We have used the inbuilt data storage of the arduino to store the data. Therefore, the number of data is limited.

When the person with the correct RFID card swipes his RFID card, his arrival time will be stored on the system using the database and an SMS will be sent the pre-given phone number confirming the student's attendance. When the same person swipes his RFID card for the second time, the system will save it as his leaving time in the database. The interval between first card swap and second card swap is the total working hours that is stored as a data.

For operating this project first user has to insert the card numbers into the microcontroller memory. It can be done by a company authority person or college administration person while issuing the card. Whenever a new student joins or a new employee is recruited in an organization/company at that time, cards will be issued. And the same entry will be made in the microcontroller program memory. In the current project, these numbers are stored in the microcontroller's program memory. Which means while burning the program into memory, we need to add these card numbers into the program. Then this card will be issued to the respective person.

The details of the student/employee has to be added to the database using the GUI. The details of the student/employee (Name, Phone no., Attendance, etc.) can be viewed and edited using the GUI. The GUI is made using Visual Studio in C# (C Sharpe).

When the user brings the RFID card near the sensor, it reads it and checks the card's phone number. It then sends the signal to the GSM module to send the SMS. Also,

the arduino sends data to the database so that the attendance is marked. This updated attendance can then be viewed on the GUI application. Same situation happens for logout.

When a student or employee has lost his/her card. Then in such a situation he/she has to report this incident to the administration person. Then, the RFID system maintenance person can clone the RFID card. The process is complicated, so the college/company authorities cannot do it on their own. It is not recommended to delete the RFID from the memory as it will require reprogramming of the Arduino.

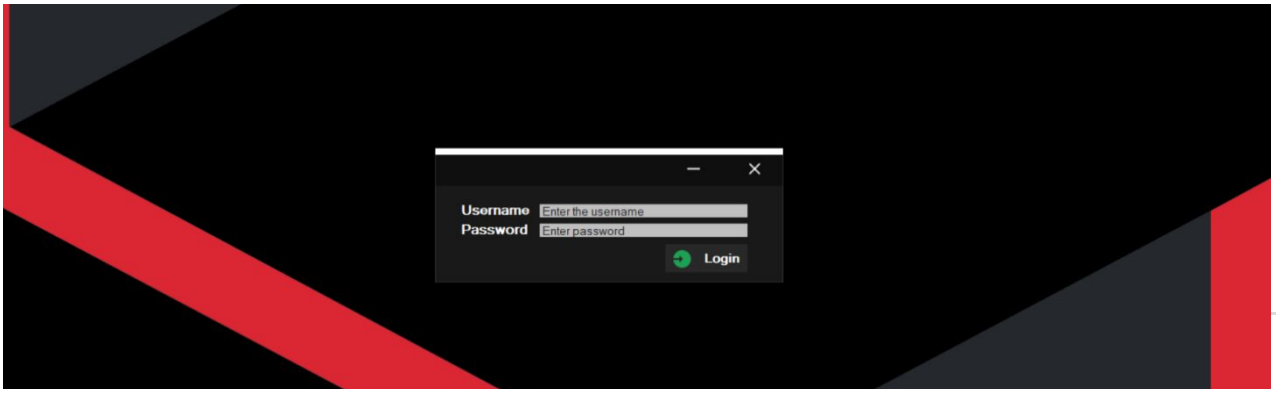
CHAPTER 10: COMPUTER INTERFACING WITH RFID BASED ATTENDANCE SYSTEM

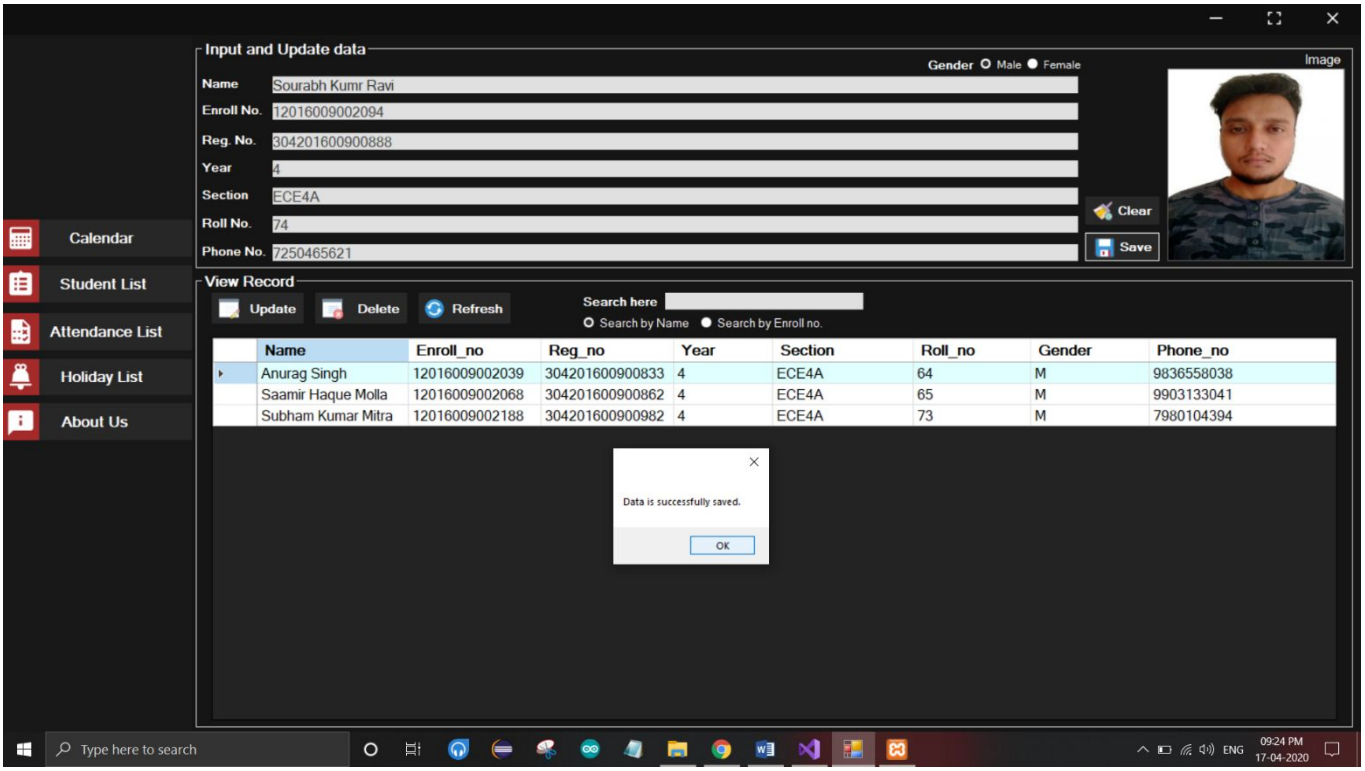
We also have a GUI for this project, so that personal details as well as attendance of the students/employees can be seen on the computer.

Data is stored in a database which can be accessed through the GUI Windows application. The GUI is designed using Visual Studio on C# (C Sharpe).

Details stored in database:

- Name
- Enrolment number
- Registration number
- Year
- Section
- Roll number
- Gender
- Phone number
- Photo
- Attendance





First a login screen appears on starting the application. The administrator has to enter the Username and Password and then has to click on Login button. If the given username and password is correct, then the person can view the details of the students/employees. If the username or password is invalid, the user is denied access. If access is allowed, the user can view, edit and delete the data of the students/employees.

CHAPTER 11: APPLICATIONS

1. This project can be used in various software companies, production industries and many other industries to take the attendance of employees. Presently many of these companies have an attendance register or muster which is a traditional and old way of maintaining attendance. We can replace it with an automated system for taking the attendance of employees.

2. It can be used in colleges, various educational institutes as well as university campuses for taking the attendance of students. Education institutes have roll-call muster or teachers take attendance and note it down manually on paper. We can replace it by our automated attendance system using RFID.

3. It can be used in shops, shopping malls for the attendance of employees and workers.

4. We can also use it to note down the in and out time of vehicles. With little bit modification, this project can be used in vehicle/car parking systems. If parking charges are charged on an hourly basis, we can use this project to note the exact in and out time of car to find out the total and exact parking charges of that particular vehicle.

CHAPTER 12: ADVANTAGES

1. This system is fully automated and it does not require any human interaction except setting the initial time setting.
2. GUI is provided with an RFID based attendance system. This gives the benefit of viewing and editing the personal as well as attendance details without the knowledge of any database language (example: SQL).
3. This system is accurate and can avoid proxy or false attendance.
4. This system also saves time whereas manual way of taking attendance is very time consuming.

CHAPTER 13: CONCLUSION & FUTURE SCOPE

13.1 CONCLUSION

The objective to build a RFID based attendance system was successfully achieved. In terms of performance and efficiency, this project has provided a convenient method of attendance marking compared to the traditional method of attendance system. By using databases, the data is more organized. This system is also a user-friendly system as data manipulation and retrieval can be done via the GUI interface, which makes it easy to use for any person. Thus, it can be implemented in either an academic institution or in organizations.

13.2 FUTURE SCOPE

1. We can make an online portal for the students as well as the administrators using which they can view and edit their personal details and can also check their attendance, thus making it easy to access the details from anywhere.
2. We can make the system compact as well as reduce the cost of production using only the microprocessor Atmega328P as we can then add the peripherals to it which are required.
3. We can add an external memory unit to it as Arduino's built in memory is limited.
4. We can use LAN to connect several RFID Readers so that the data can be uploaded live on the main server host.

PROGRAMMING CODES FOR ARDUINO

1. MFRC522 CODE FOR GETTING RFID ID NUMBERS

```
* Typical pin layout used:
 * -----------------------------------------------------------------------------------------------------
 *             MFRC522      Arduino       Arduino   Arduino    Arduino            Arduino
 *             Reader/PCD   Uno/101       Mega      Nano v3    Leonardo/Micro     Pro Micro
 * Signal      Pin          Pin           Pin       Pin        Pin                Pin
* -----------------------------------------------------------------------------------------------------
 * RST/Reset   RST          9             5         D9         RESET/ICSP-5       RST
 * SPI SS      SDA(SS)      10            53        D10        10                 10
 * SPI MOSI    MOSI         11 / ICSP-4   51        D11        ICSP-4             16
 * SPI MISO    MISO         12 / ICSP-1   50        D12        ICSP-1             14
 * SPI SCK     SCK          13 / ICSP-3   52        D13        ICSP-3             15
 */

#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN         9          // Configurable, see typical pin layout above
#define SS_PIN          10         // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance
```

```
void setup() {
	Serial.begin(9600);		// Initialize serial communications with the PC
	while (!Serial);		// Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)
	SPI.begin();			// Init SPI bus
	mfrc522.PCD_Init();		// Init MFRC522
	delay(4);				// Optional delay. Some board do need more time after init to be ready, see Readme
	mfrc522.PCD_DumpVersionToSerial();	// Show details of PCD - MFRC522 Card Reader details
	Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));
}

void loop() {
	// Reset the loop if no new card present on the sensor/reader. This saves the entire process when idle.
	if ( ! mfrc522.PICC_IsNewCardPresent()) {
		return;
	}
	// Select one of the cards
	if ( ! mfrc522.PICC_ReadCardSerial()) {
		return;
	}
// Dump debug info about the card; PICC_HaltA() is automatically called
```

```
        mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
```

2. ARDUINO CODE FOR DETECTING AND SENDING SMS

```
* Typical pin layout used:
* -----------------------------------------------------------------------------------------
 *   Signal                Pin                Pin                Pin
 *                         Arduino Uno        Arduino Mega       MFRC522 board
* -----------------------------------------------------------------------------------------
*    Reset                9                  5                  RST
 * SPI SS                10                 53                 SDA
 * SPI MOSI              11                 51                 MOSI
 * SPI MISO              12                 50                 MISO
 * SPI SCK               13                 52                 SCK
 * Voltage 3.3v
 */

 // gsm module is connected with pin number 7 and pin number 8 of the arduino.

#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8); // gsm module connected here
```

```
String textForSMS;



#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);        // Create MFRC522 instance.

// lcd pins
#define rs A1
#define en A2
#define d4 6
#define d5 5
#define d6 4
#define d7 3
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// parents numbers

String f1001 = "+923339218213"; // student1 father cell phone number
String f1002 = "+923351920959";
String f1003 = "+923171956677";

void setup() {
```

```
        Serial.begin(9600);        // Nodemcu is connected over here
        randomSeed(analogRead(0));
        SIM900.begin(9600); // original 19200. while enter 9600 for sim900A
        SPI.begin();               // Init SPI bus

mfrc522.PCD_Init();        // Init MFRC522 card
        //Serial.println("Scan a MIFARE Classic PICC to demonstrate Value Blocks.");

          // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("AttendanceSystem");
  delay(3000);
  lcd.clear();
  lcd.print("Swip Your Card");
  delay(1000);
}

void loop() {


        // Prepare key - all keys are set to FFFFFFFFFFFFh at chip delivery from the factory.
        MFRC522::MIFARE_Key key;
        for (byte i = 0; i < 6; i++) {
                key.keyByte[i] = 0xFF;
```

```
	}
	// Look for new cards
	if ( ! mfrc522.PICC_IsNewCardPresent()) {
		return;
	}

	// Select one of the cards
	if ( ! mfrc522.PICC_ReadCardSerial()) {
		return;
	}
	// Now a card is selected. The UID and SAK is in mfrc522.uid.

	// Dump UID
	Serial.print("Card UID:");
	for (byte i = 0; i < mfrc522.uid.size; i++) {
	    //  Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
	    //  Serial.print(mfrc522.uid.uidByte[i], DEC);
	}
	Serial.println();

	// Dump PICC type
	byte piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
   //   Serial.print("PICC type: ");
//Serial.println(mfrc522.PICC_GetTypeName(piccType));
	if (	piccType != MFRC522::PICC_TYPE_MIFARE_MINI
```

```
        &&      piccType != MFRC522::PICC_TYPE_MIFARE_1K
        &&      piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
        //Serial.println("This sample only works with MIFARE Classic cards.");
        return;
    }


// defining Cards here


        // student1
      if( (mfrc522.uid.uidByte[0] == 224) && (mfrc522.uid.uidByte[1] == 154) && (mfrc522.uid.uidByte[2] == 156) && (mfrc522.uid.uidByte[3] == 124) ) // student1 card
    {
     lcd.clear();
     lcd.print("AttendanceMarked");
     delay(500);
Serial.println("student 1");


    // for gsm
     sendsms(" Ali is Present", f1001);
     delay(1000);


    lcd.setCursor(0, 1);
    lcd.print("message sent");
    delay(2000);
```

```
        lcd.clear();
         lcd.print("Swip Your Card");
         delay(1000);
        }



 // student2
            if( (mfrc522.uid.uidByte[0] == 189) && (mfrc522.uid.uidByte[1] == 54) &&
(mfrc522.uid.uidByte[2] == 235) && (mfrc522.uid.uidByte[3] == 213) ) // student2 card
        {
        lcd.clear();
         lcd.print("AttendanceMarked");
         delay(500);
  Serial.println("student 2");

        // for gsm
         sendsms("James is Present", f1002);
         delay(1000);

        lcd.setCursor(0, 1);
        lcd.print("message sent");
        delay(2000);
```

```
          lcd.clear();
           lcd.print("Swip Your Card");
           delay(1000);
         }
        else
        Serial.println("unregistered user");
       }



void sendsms(String message, String number)
{
String mnumber = "AT + CMGS = \""+number+"\"";
   SIM900.print("AT+CMGF=1\r");
  delay(1000);
 SIM900.println(mnumber);  // recipient's mobile number, in international format


  delay(1000);
  SIM900.println(message);                    // message to send
  delay(1000);
  SIM900.println((char)26);                   // End AT command with a ^Z, ASCII code 26
  delay(1000);
  SIM900.println();
  delay(100);                                 // give module time to send SMS
 // SIM900power();
}
```

3. VISUAL STUDIO CODE FOR_LOGIN (C SHARPE)

```
using System;
using System.Windows.Forms;
using Xamarin.Forms.PlatformConfiguration;
using MySql.Data.MySqlClient;
using System.Drawing;
using System.Configuration;
using System.Data.SqlClient;
using System.Data;
using System.Runtime.InteropServices;
using System.Windows.Media;
using System.IO;
namespace Project_ECE
{
   public partial class Login : Form
   {
      public Login()
      {
         InitializeComponent();
         //Form
         this.Text = string.Empty;
         this.ControlBox = false;
         this.DoubleBuffered = true;
```

```
    }

    [DllImport("user32.DLL", EntryPoint = "ReleaseCapture")]
    private extern static void ReleaseCapture();
    [DllImport("user32.DLL", EntryPoint = "SendMessage")]
     private extern static void SendMessage(System.IntPtr hWnd, int wMsg, int wParam, int lParam);

    private void Btn_Min_Click(object sender, EventArgs e)
    {
       this.WindowState = FormWindowState.Minimized;
    }

    private void Btn_Login_Click(object sender, EventArgs e)
    {
       try
       {
                                                    string constring = "Server=localhost;Database=attendance_system;Uid=root;Pid=";
            MySqlConnection conn = new MySqlConnection(constring);

                    MySqlCommand cmd = new MySqlCommand("SELECT * FROM ATTENDANCE_SYSTEM.USER_PASS WHERE Username=LTRIM('" + this.TextBoxUsrNm.Text + "') AND Password=LTRIM('" + this.TextBoxPass.Text + "');", conn);
            MySqlDataReader rdr;
            conn.Open();
```

```
            rdr = cmd.ExecuteReader();
            int count = 0;

while (rdr.Read())
            {
               count += 1;
            }
            if (count == 1)
            {
                MessageBox.Show(" Welcome Username: " + this.TextBoxUsrNm.Text +
".");
               this.Hide();
               Form1 f1 = new Form1();
               f1.ShowDialog();
               this.Close();
            }
            else
            {
               MessageBox.Show("Invalid Username or Password.");
               conn.Close();
            }
         }
         catch (Exception ex)
         {
            MessageBox.Show(ex.Message);
```

```
        }
    }


private void Btn_Close_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void Pnl_Top_Paint(object sender, PaintEventArgs e)
    {


    }

    private void Pnl_Top_MouseDown(object sender, MouseEventArgs e)
    {
        ReleaseCapture();
        SendMessage(this.Handle, 0x112, 0xf012, 0);
    }

    private void Btn_Close_MouseMove(object sender, MouseEventArgs e)
    {
        Btn_Close.BackColor = System.Drawing.Color.FromArgb(130, 0, 10);
    }
```

```
    private void Btn_Min_MouseMove(object sender, MouseEventArgs e)
    {
        Btn_Min.BackColor = System.Drawing.Color.FromArgb(27, 99, 58);
    }

    private void Btn_Min_MouseLeave(object sender, EventArgs e)
    {
        Btn_Min.BackColor = System.Drawing.Color.FromArgb(15, 15, 15);
    }

    private void Btn_Close_MouseLeave(object sender, EventArgs e)
    {
        Btn_Close.BackColor = System.Drawing.Color.FromArgb(15, 15, 15);
    }

    private void Btn_Login_MouseMove(object sender, MouseEventArgs e)
    {
        Btn_Login.BackColor = System.Drawing.Color.FromArgb(50, 50, 50);
    }

    private void Btn_Login_MouseLeave(object sender, EventArgs e)
    {
        Btn_Login.BackColor = System.Drawing.Color.FromArgb(40, 40, 40);
    }
}
```

```
}
```

4. VISUAL STUDIO CODE FOR MAIN APPLICATION (C SHARPE)

```
using System;
using System.Windows.Forms;
using Xamarin.Forms.PlatformConfiguration;
using MySql.Data.MySqlClient;
using System.Drawing;
using System.Configuration;
using System.Data.SqlClient;
using System.Data;
using System.Runtime.InteropServices;
using System.Windows.Media;
using System.IO;

namespace Project_ECE
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
         this.WindowState = FormWindowState.Normal;
         LoadDataIntoDataGridView();
         //Form
```

```
        this.Text = string.Empty;

       this.ControlBox = false;
        this.DoubleBuffered = true;
        this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;
    }

    //variable for student identification
    private string Student_identification;

    //variable for DataTable
    DataTable dtRecords;

    private void LoadDataIntoDataGridView() //show records
    {
        SqlConnection conn = new SqlConnection(connection_string());

                                        string query = "SELECT * FROM
ATTENDANCE_SYSTEM.STUDENT_DETAILS;";
        SqlDataAdapter sda = new SqlDataAdapter(query, conn);

        DataTable data = new DataTable();
        sda.Fill(data);
        DataGridViewImageColumn dgv = new DataGridViewImageColumn();
        dgv = (DataGridViewImageColumn)DataGridView1.Columns[8];
```

```
            dgv.ImageLayout = DataGridViewImageCellLayout.Stretch;
            DataGridView1.DataSource = data;

        }

    //connect database
    public string connection_string()
    {
                                                    string constring = "Server=localhost;Database=attendance_system;Uid=root;Pid=";
            return constring;
    }



    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void Btn_Std_Dets_Click(object sender, EventArgs e)
    {

    }
```

```
private void button1_Click(object sender, EventArgs e)
{



  }

private void label1_Click(object sender, EventArgs e)
{

}




private bool IsValid()
{
   if (TextBoxName.Text.Trim() == "Enter full name")
   {
      MessageBox.Show("Student Name is required.", "Required field error.");
      return false;
   }
   else if (TextBoxEnroll.Text.Trim() == string.Empty)
   {
         MessageBox.Show("Student Enrollment No. is required.", "Required field error.");
```

```
            return false;
        }
        else if (TextBoxRegNo.Text.Trim() == string.Empty)
        {

MessageBox.Show("Student Registration No. is required.", "Required field error.");
            return false;
        }
        else if (TextBoxYear.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Student Year is required.", "Required field error.");
            return false;
        }
        else if (TextBoxSection.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Student Section is required.", "Required field error.");
            return false;
        }
        else if (TextBoxRoll.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Student Roll No. is required.", "Required field error.");
            return false;
        }
        else if (TextBoxPhNo.Text.Trim() == string.Empty)
        {
```

```
            MessageBox.Show("Student Phone No. is required.", "Required field error.");
            return false;
        }
        else if (RadBtn_Male.Checked == false && Rad_Btn_Female.Checked == false)
        {

  MessageBox.Show("Student Gender is required.", "Required selection error.");
            return false;
        }
        else if (PictureBoxImageInput.Image == null)
        {
                MessageBox.Show("Student Image is required.", "Required image upload
error.");
            return false;
        }
        else
            return true;
    }


    private void Btn_Calendar_Click(object sender, EventArgs e)
    {

    }
```

```
//Drag form
[DllImport("user32.DLL", EntryPoint = "ReleaseCapture")]
private extern static void ReleaseCapture();
[DllImport("user32.DLL", EntryPoint = "SendMessage")]
 private extern static void SendMessage(System.IntPtr hWnd, int wMsg, int wParam, int lParam);



private void Pnl_Top_MouseDown(object sender, MouseEventArgs e)
{
   ReleaseCapture();
   SendMessage(this.Handle, 0x112, 0xf012, 0);
}


 private void TextBoxSearch_TextChanged(object sender, EventArgs e) //search text box
{
   DataView DataShow = new DataView(dtRecords);


   if (RadBtn_SrchName.Checked)
   {
                  DataShow.RowFilter = string.Format("NAME LIKE '%{0}%'", TextBoxSearch.Text); //Search by Name
      DataGridView1.DataSource = DataShow;
   }
   else
```

```
        {
                DataShow.RowFilter = string.Format("ENROLL_NO LIKE '%{0}%'",
TextBoxSearch.Text); //Search by Enroll_no
            DataGridView1.DataSource = DataShow;
        }
    }

    private void radioButton1_CheckedChanged(object sender, EventArgs e)

        {

    }

    private void Pnl_Top_Paint(object sender, PaintEventArgs e)
    {

    }

    private void TextBoxName_TextChanged_1(object sender, EventArgs e)
    {

    }

    public void TextBoxName_Enter(object sender, EventArgs e)
    {
```

```
        if (TextBoxName.Text == " Enter full name")
        {
            TextBoxName.Text = " ";
            TextBoxName.ForeColor = System.Drawing.Color.FromArgb(40, 40, 40);
        }
    }

    public void TextBoxName_Leave(object sender, EventArgs e)
    {

if (TextBoxName.Text == " ")
        {
            TextBoxName.Text = " Enter full name";
            TextBoxName.ForeColor = System.Drawing.Color.Silver;
        }
    }

    private void ButtonClearAll_MouseMove(object sender, MouseEventArgs e)
    {
        ButtonClearAll.BackColor = System.Drawing.Color.FromArgb(50, 50, 50);
    }

    private void ButtonClearAll_MouseLeave(object sender, EventArgs e)
    {
        ButtonClearAll.BackColor = System.Drawing.Color.FromArgb(40, 40, 40);
```

```
    }


    //MINIMIZE BUTTON
    //for colour change
    private void Btn_Min_MouseMove(object sender, MouseEventArgs e)
    {
        Btn_Min.BackColor = System.Drawing.Color.FromArgb(27, 99, 58);
    }


private void Btn_Min_MouseLeave(object sender, EventArgs e)
    {
        Btn_Min.BackColor = System.Drawing.Color.FromArgb(15, 15, 15);
    }

    //for button click event
    private void Btn_Min_Click(object sender, EventArgs e)
    {
        this.WindowState = FormWindowState.Minimized;
    }

    //CLOSE BUTTON
    //for colour change
    private void Btn_Close_MouseLeave(object sender, EventArgs e)
```

```
    {
        Btn_Close.BackColor = System.Drawing.Color.FromArgb(15, 15, 15);
    }

    private void Btn_Close_MouseDown(object sender, MouseEventArgs e)
    {
        Btn_Close.BackColor = System.Drawing.Color.FromArgb(100, 0, 10);
    }

    private void Btn_Close_MouseMove_1(object sender, MouseEventArgs e)
    {

Btn_Close.BackColor = System.Drawing.Color.FromArgb(130, 0, 10);
    }

    //for button click event
    private void Btn_Close_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    //MAXIMIZE BUTTON
    //for colour change
    private void Btn_Max_MouseMove(object sender, MouseEventArgs e)
    {
```

```
    Btn_Max.BackColor = System.Drawing.Color.FromArgb(158, 69, 0);
}

private void Btn_Max_MouseLeave(object sender, EventArgs e)
{
    Btn_Max.BackColor = System.Drawing.Color.FromArgb(15, 15, 15);
}

//for button click event
private void Btn_Max_Click(object sender, EventArgs e)
{
    if (this.WindowState.Equals(FormWindowState.Normal))

    {
        this.WindowState = FormWindowState.Maximized;
    }
    else
    {
        this.WindowState = FormWindowState.Normal;
    }
}

//save button
private void ButtonSave_MouseMove(object sender, MouseEventArgs e)
{
```

```
    ButtonSave.BackColor = System.Drawing.Color.FromArgb(50, 50, 50);
}

private void ButtonSave_MouseLeave(object sender, EventArgs e)
{
    ButtonSave.BackColor = System.Drawing.Color.FromArgb(40, 40, 40);
}

//edit button
private void ButtonEdit_MouseMove(object sender, MouseEventArgs e)
{
    ButtonUpdate.BackColor = System.Drawing.Color.FromArgb(50, 50, 50);
}

private void ButtonEdit_MouseLeave(object sender, EventArgs e)
{
    ButtonUpdate.BackColor = System.Drawing.Color.FromArgb(40, 40, 40);
}

//delete button
private void ButtonDelete_MouseMove(object sender, MouseEventArgs e)
{
    ButtonDelete.BackColor = System.Drawing.Color.FromArgb(50, 50, 50);
}
```

```
    private void ButtonDelete_MouseLeave(object sender, EventArgs e)
    {
        ButtonDelete.BackColor = System.Drawing.Color.FromArgb(40, 40, 40);
    }

    //refresh button
    private void ButtonRefresh_MouseMove(object sender, MouseEventArgs e)
    {
        ButtonRefresh.BackColor = System.Drawing.Color.FromArgb(50, 50, 50);
    }

    private void ButtonRefresh_MouseLeave(object sender, EventArgs e)
    {
        ButtonRefresh.BackColor = System.Drawing.Color.FromArgb(40, 40, 40);

}

    //datagrid
                    private void DataGridView1_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
    {
                                                    Student_identification = Convert.ToString(DataGridView1.SelectedRows[0].Cells[0].Value);
        TextBoxName_Enter(sender, e);
                                                    TextBoxName.Text = DataGridView1.SelectedRows[0].Cells[0].Value.ToString();
```

```
                                                    TextBoxEnroll.Text =
DataGridView1.SelectedRows[0].Cells[1].Value.ToString();
                                                    TextBoxRegNo.Text =
DataGridView1.SelectedRows[0].Cells[2].Value.ToString();
        TextBoxYear.Text = DataGridView1.SelectedRows[0].Cells[3].Value.ToString();
                                                    TextBoxSection.Text =
DataGridView1.SelectedRows[0].Cells[4].Value.ToString();
        TextBoxRoll.Text = DataGridView1.SelectedRows[0].Cells[5].Value.ToString();
        TextBoxPhNo.Text = DataGridView1.SelectedRows[0].Cells[7].Value.ToString();

        if(DataGridView1.SelectedRows[0].Cells[6].Value.ToString() == "M")
        {
           RadBtn_Male.Checked = true;
        }
        else
        {

Rad_Btn_Female.Checked = false;
        }



    }

    private void RadBtn_Male_CheckedChanged(object sender, EventArgs e)
    {
```

```
    }

    private void Rad_Btn_Female_CheckedChanged(object sender, EventArgs e)
    {

    }


    //SAVE BUTTON
    private void ButtonSave_Click(object sender, EventArgs e)
    {
       if (IsValid())
       {



try
          {
             //connection to database
             MySqlConnection conn = new MySqlConnection(connection_string());

             //set gender value
             string gender = string.Empty;
             if (RadBtn_Male.Checked)
```

```
{
   gender = "M";
}
else
{
   gender = "F";
}

//add data into database
conn.Open();
MySqlCommand cmd;

cmd = conn.CreateCommand();
                                    cmd.CommandText = "INSERT INTO
ATTENDANCE_SYSTEM.STUDENT_DETAILS
(Name,Enroll_no,Reg_no,Year,Section,Roll_no,Gender,Phone_no,Image)
VALUES(LTRIM(@Name),@Enroll_no,@Reg_no,@Year,@Section,@Roll_no,@Gende
r,@Phone_no,@Image)";

cmd.Parameters.AddWithValue("@Name", TextBoxName.Text);
cmd.Parameters.AddWithValue("@Enroll_no", TextBoxEnroll.Text);
cmd.Parameters.AddWithValue("@Reg_no", TextBoxRegNo.Text);
cmd.Parameters.AddWithValue("@Year", TextBoxYear.Text);
cmd.Parameters.AddWithValue("@Section", TextBoxSection.Text);
cmd.Parameters.AddWithValue("@Roll_no", TextBoxRoll.Text);
cmd.Parameters.AddWithValue("@Gender", gender);
```

```
            cmd.Parameters.AddWithValue("@Phone_no", TextBoxPhNo.Text);
            cmd.Parameters.Add(new MySqlParameter("@Image", SavePhoto()));

            cmd.ExecuteNonQuery();

            conn.Close();

            MessageBox.Show("Data is successfully saved.");
        }
        catch (Exception e_save)
        {
            MessageBox.Show(e_save.Message);
        }
        finally
        {
            LoadDataIntoDataGridView();
        }
    }

}

  public byte[] SavePhoto()
  {
     MemoryStream mstream = new MemoryStream();
```

```
                                        PictureBoxImageInput.Image.Save(mstream,
PictureBoxImageInput.Image.RawFormat);
            return mstream.GetBuffer();
        }

        //clear button
        private void ButtonClearAll_Click(object sender, EventArgs e)
        {
            TextBoxName.Clear();
            TextBoxName_Leave(sender, e);
            TextBoxEnroll.Clear();
            TextBoxEnroll.Clear();
            TextBoxRegNo.Clear();
            TextBoxYear.Clear();
            TextBoxSection.Clear();
            TextBoxRoll.Clear();
            TextBoxPhNo.Clear();
            PictureBoxImageInput.Image = null;

        }

        private void ButtonUpdate_Click(object sender, EventArgs e)
        {
            if (IsValid())
            {
```

```
try
{
   //connection to database
   MySqlConnection conn = new MySqlConnection(connection_string());

   //set gender value
   string gender = string.Empty;
   if (RadBtn_Male.Checked)
   {
      gender = "M";
   }
   else
   {
      gender = "F";
   }

   //update data in database
   conn.Open();
   MySqlCommand cmd;

   cmd = conn.CreateCommand();
                                        cmd.CommandText = "UPDATE
ATTENDANCE_SYSTEM.STUDENT_DETAILS                    SET
(Name=LTRIM(@Name),Enroll_no=@Enroll_no,Reg_no=@Reg_no,Year=@Year,Secti
```

```
on=@Section,Roll_no,=@Roll_no,Gender=@Gender,Phone_no=@Phone_no,Image=@I
mage);";
                cmd.Parameters.AddWithValue("@Name", TextBoxName.Text);
                cmd.Parameters.AddWithValue("@Enroll_no", TextBoxEnroll.Text);
                cmd.Parameters.AddWithValue("@Reg_no", TextBoxRegNo.Text);
                cmd.Parameters.AddWithValue("@Year", TextBoxYear.Text);
                cmd.Parameters.AddWithValue("@Section", TextBoxSection.Text);
                cmd.Parameters.AddWithValue("@Roll_no", TextBoxRoll.Text);
                cmd.Parameters.AddWithValue("@Gender", gender);
                cmd.Parameters.AddWithValue("@Phone_no", TextBoxPhNo.Text);
                cmd.Parameters.Add(new MySqlParameter("@Image", SavePhoto()));

                cmd.ExecuteNonQuery();

                conn.Close();

                MessageBox.Show("Data is successfully updated.");
            }
            catch (Exception e_save)
            {
                MessageBox.Show(e_save.Message);
            }

        finally
            {
```

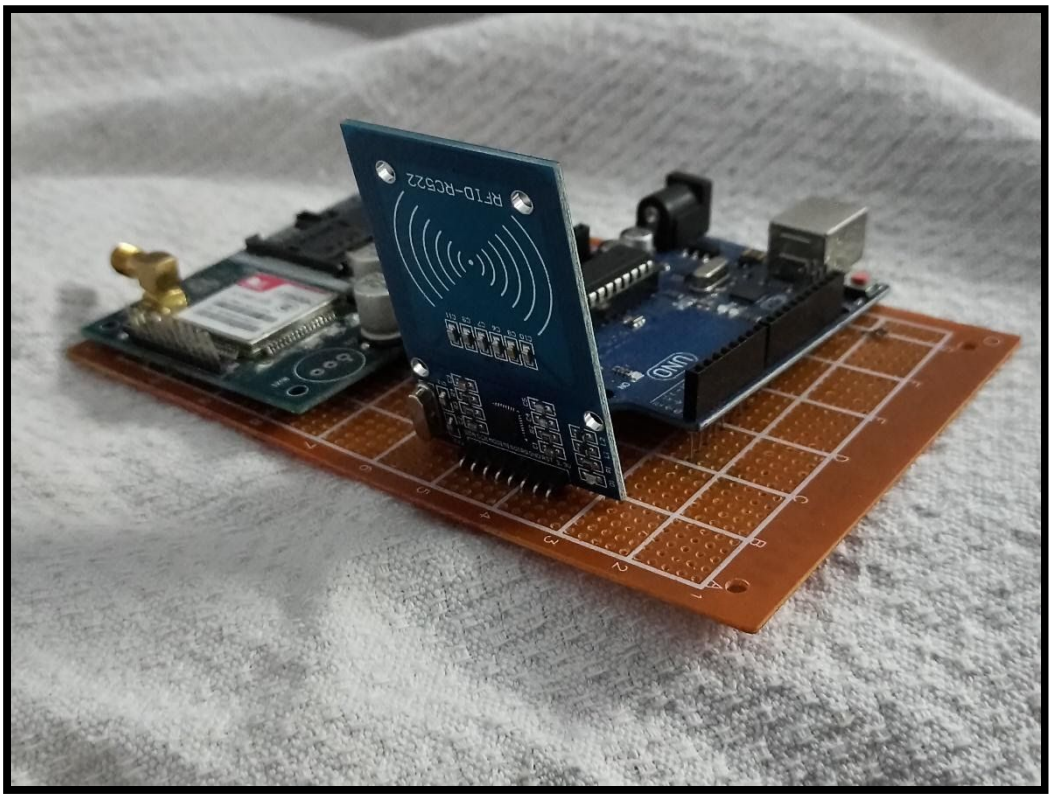
```
            LoadDataIntoDataGridView();
        }
    }
}

private void ButtonRefresh_Click(object sender, EventArgs e)
{
    LoadDataIntoDataGridView();
}

private void ButtonDelete_Click(object sender, EventArgs e)
{
    try
    {
            if (TextBoxEnroll.Text != " " || TextBoxEnroll.Text != " Enter Student Enrollment no")
        {
            MySqlConnection conn = new MySqlConnection(connection_string());

            //for reading radio button value
            string gender = string.Empty;
            if (RadBtn_Male.Checked)

{
```

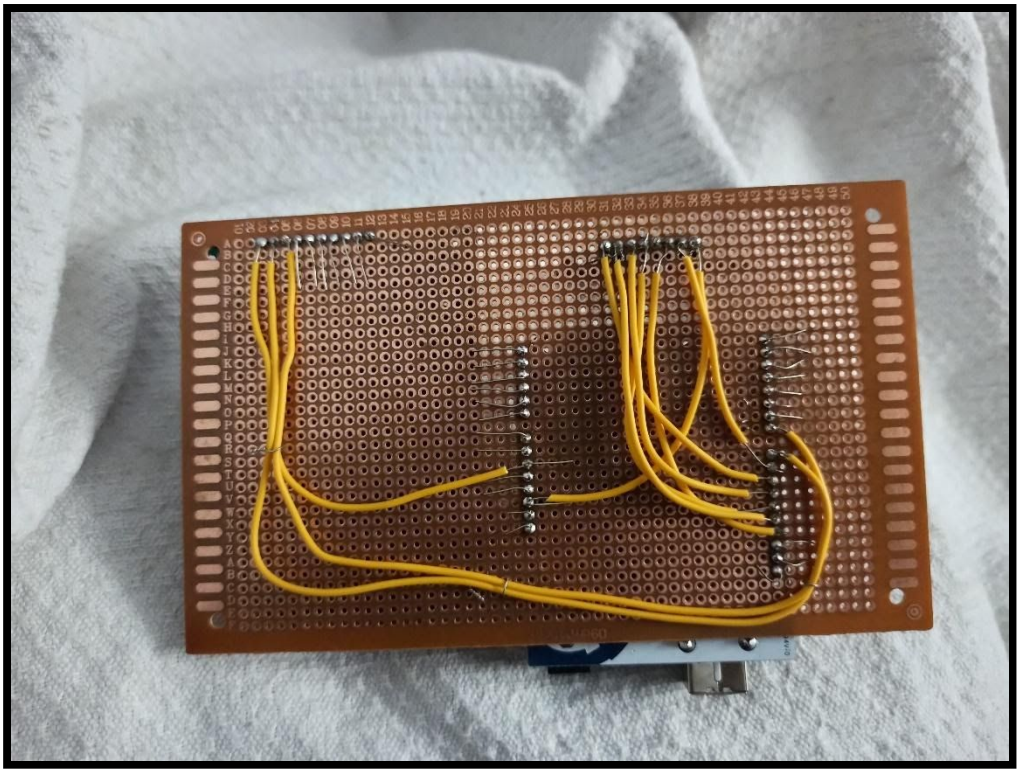
```
            gender = "M";
        }
        else
            gender = "F";

        //add data into database
        conn.Open();
        MySqlCommand cmd;

        cmd = conn.CreateCommand();
                                    cmd.CommandText = "DELETE FROM
ATTENDANCE_SYSTEM.STUDENT_DETAILS WHERE Enroll_no=@Enroll_no;";
        cmd.Parameters.AddWithValue("@Enroll_no", TextBoxEnroll.Text);

        cmd.ExecuteNonQuery();

        conn.Close();
      }
    }
    catch (Exception e1)
    {
      MessageBox.Show(e1.Message);
    }
    finally
```

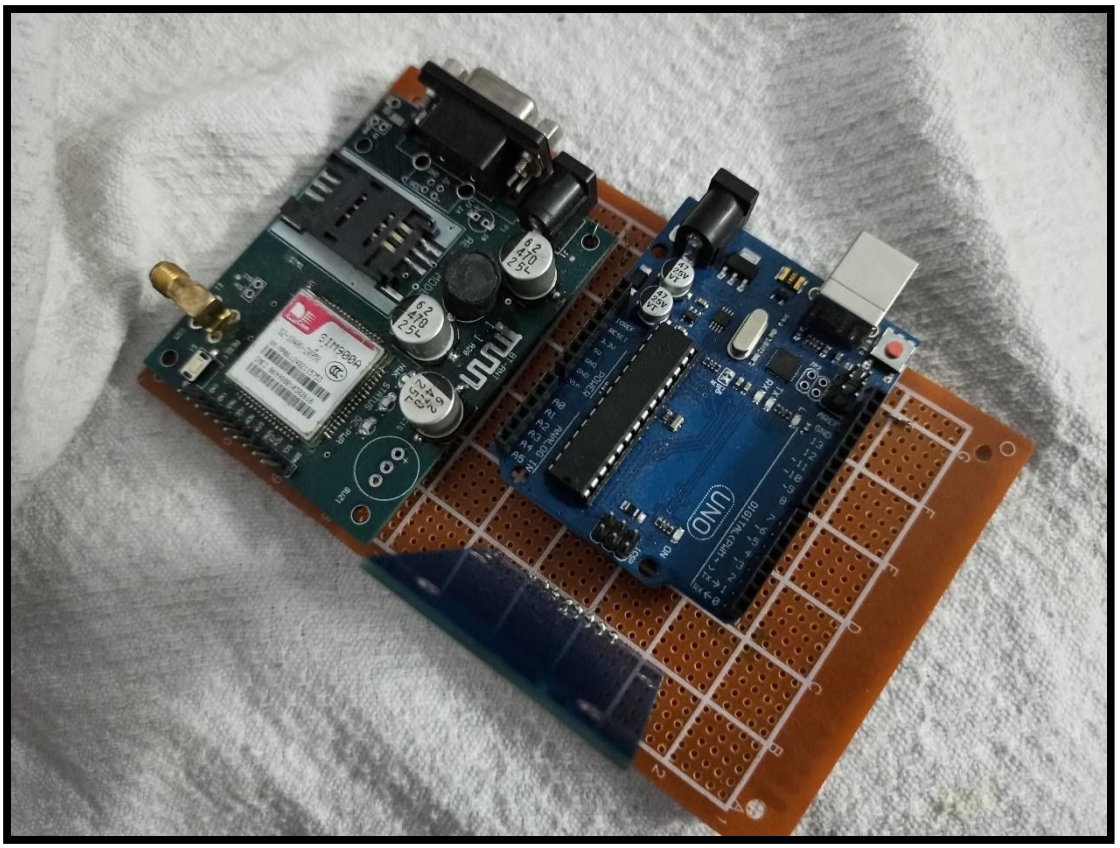
```
{
            LoadDataIntoDataGridView();
        }
    }

    private void PictureBoxImageInput_Click(object sender, EventArgs e)
    {
        try
        {
            OpenFileDialog dlg = new OpenFileDialog();
            dlg.Title = "Select Image - File size less than 64KB";
            dlg.Filter = "JPG File(*.jpg)|*.jpg";

            if (dlg.ShowDialog() == DialogResult.OK)
            {
                PictureBoxImageInput.Image = new Bitmap(dlg.FileName);
            }
        }
        catch(Exception e_img)
        {
            MessageBox.Show(e_img.Message);
        }
        finally
        {
            LoadDataIntoDataGridView();
```

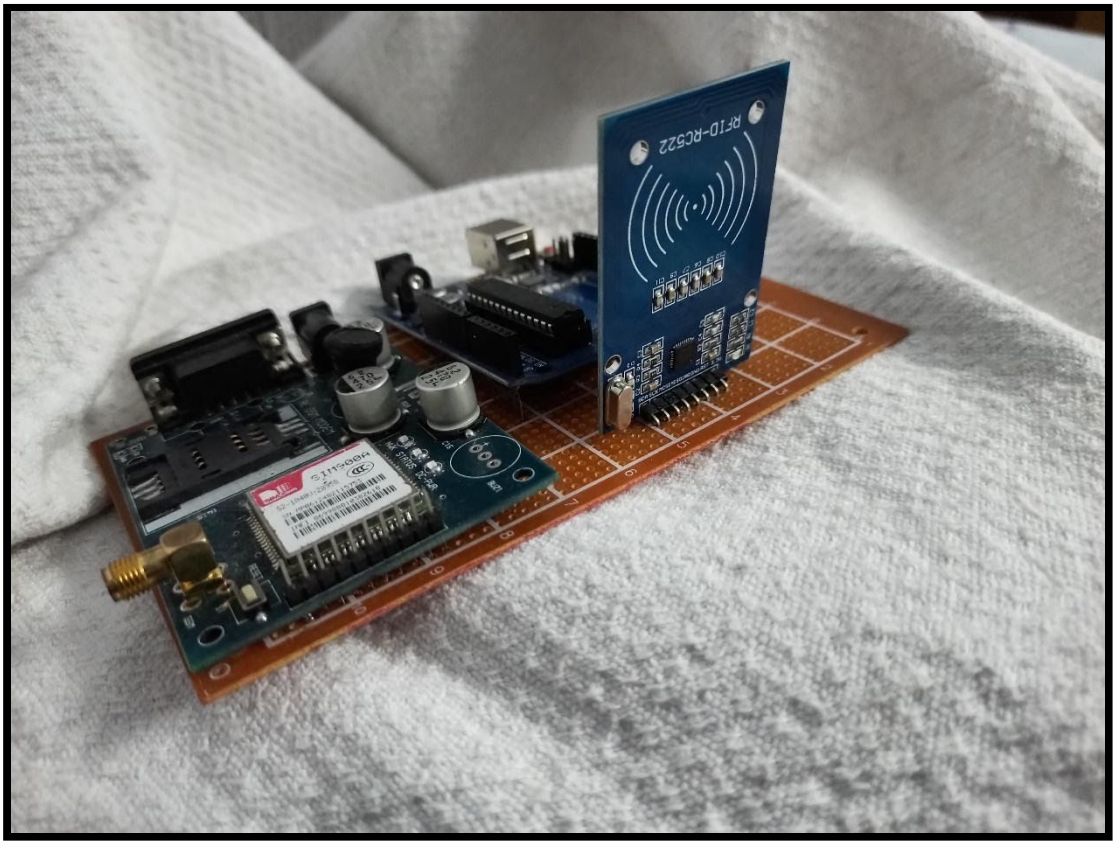
```
        }
      }
    }
}
```

SNAPSHOTS OF PROJECT MODEL








RFID-RC522

BIBLIOGRAPHY

1. www.wikipedia.org
2. https://www.youtube.com/
3. www.semanticscholar.org
4. https://www.tutorialspoint.com/arduino/arduino_board_description.htm
5. https://components101.com/buzzer-pinout-working-datasheet
6. http://wilsonhurd.com/images/pdfs/MembraneSwitchesDesignGuidelines.pdf
7. https://en.wikipedia.org/wiki/Radio-frequency_identification
8. http://www.abr.com/what-is-rfid-how-does-rfid-work
9. https://www.elprocus.com/rfid-basic-introduction-simple-application
10. http://www.instructables.com/id/Interfacing-20x4-LCD-with-Arduino
11. http://www.circuitstoday.com/a-note-on-character-lcd-displays
12. http://www.8051projects.net/lcd-interfacing/basics.php