College of Engineering

Department of Electrical Engineering

Spring 2019-2020

Senior Design Project Report

RFID Based Child Security System

In partial fulfillment of the requirements for the Degree of Bachelor of Science in Electrical Engineering

Team Members

	Student Name	Student ID
1	Abdulaziz Aljabr	201600140
2	Abdulrahman Baha	201600224
3	Abduliaziz Aldossary	201401219
4	Abduliteef Alhamdan	201300803

Project Advisors:

Advisor Name: Mr. Ahmed Abul Hussain

Abstract

Nowadays schools have a large number of students, and it is hard to keep track and ensure the safety of all students manually. School faculty are under a lot of pressure form students' parents to ensure their kids safety. Many accidents occurred to students being left out on the bus and died from suffocation or students being lost and not arrive to school. We know how much parents value their children; therefore, our project proposes a solution to these problems using RFID, GSM, and GPS technologies.

Using RFID technology will automate the process of monitoring students on the bus and taking attendance leaving very little for error which will offer more safety to students and save time. The aim of this project is to create a school system based on RFID to monitor students entering and leaving the bus and assure no one is left on the bus before it is locked and send SMS message to the parents once their child has arrived to school. Also, GPS will be placed on the bus to monitor the speed of the bus and send an SMS message if the legal speed was exceeded. There will be also a user interface for the bus driver to be able to monitor the students riding the bus, and alarming when there is a student still on the bus. Once students arrive to class, their attendance will be taken and will be saved in a database. Parents will be able to see their student's location on the bus and their student's attendance on a graphical user interface.

Table of *Contents*

1. Introduction

1.1 Project Definition

The project purpose is to design an RFID-based safe transportation and attendance monitoring system for pre-elementary and elementary school going children.

1.2 Project Objectives

- To assure the safety of young students to be delivered to their destination.
- To save time with more efficiency of taking the attendance
- To learn about RFID technology and its application in solving real world problem.
- To increase awareness of safety issues.

1.3 Project Specifications

- RFID tags can be detected within range of at least 5 meters.
- SMS message with dates and exact time to parents for attendance in school
- SMS message with dates and exact time to parents for student riding the bus
- Keep a student record of attendance.
- Notify the bus driver if a student did not leave the bus.
- Using GSM to send SMS
- RFID reader to detect tags

1.4 Product Architecture and Components

The project architecture shown in Figure 1 explains the flow of the project operation. In dark orange are the four sub-systems. The bus, class, and user interface all communicates with the database through Wi-Fi. The database will be the collection of the information using MySQL at the school. The sub-system user interface in the project architecture is aimed to be for students' parents. The bus, and the class have sub-sub-systems. The bus will have four main sub-sub-systems which are: RFID reader, user interface, GPS, and GSM. These sub-sub-systems are going to serve the purposes of ensuring the safety of the students. It will take in and out students with exact time using the RFID reader. It will use GPS to measure speed in order to find if the driver is speeding. It will have driver user interface with Wi-Fi connection for the driver to be able to monitor the students. It will use GSM to send messages if the bus is speeding or notify parents with students' entrance and leaving the bus. The class will also have two sub-subsystems which are: the RFID reader, and GSM. The RFID reader will read the student to determine if the student is late, attended, or absent. Then the GSM will be sending SMS if the student is absent. Using Wi-Fi, it will record all of the information to the database. The project architecture explains the flow in a simple and clear way.

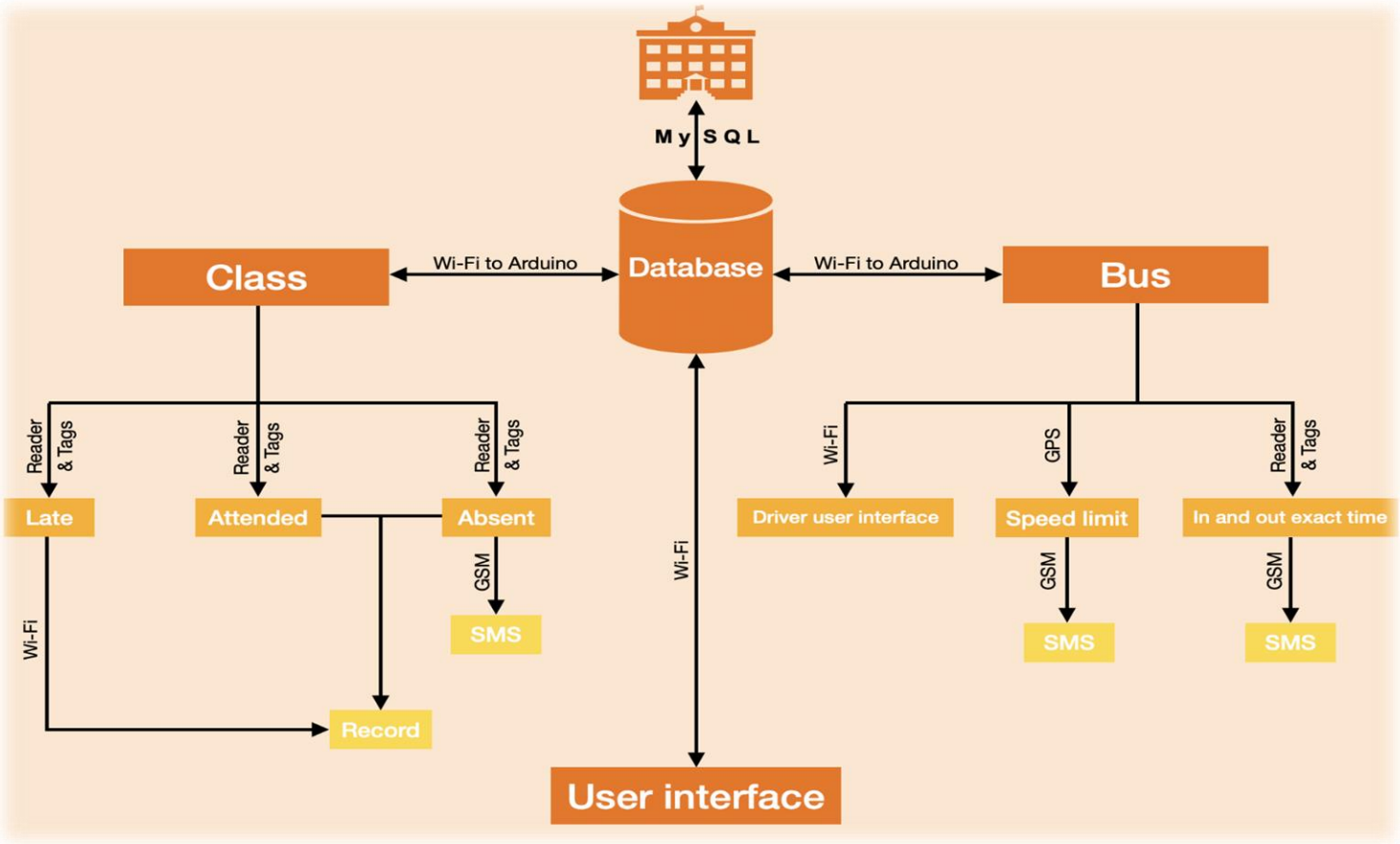


Figure 1: Project Architecture

Project Components:

- Two UHF RFID readers
- 20 RFID tags
- Two GSM shields
- GPS
- Arduino Mega2560 and Arduino due
- Wi-Fi ESP32
- NodeMCU

1.5 Applications

The project is RFID based and could be implemented to serve other application. Our main applications are:

- Track students in and out of the bus with alarming when a student is still in the bus.
- Attendance system for students.

With small adjustments, the project will also be able to serve other applications such as:

- Employees attendance in any company.
- Employees bus monitoring.
- Inventory monitoring.

Our considerations in the project and system might not work best with other application but it would be applicable if small adjustments were made.

2. Literature Review

2.1 Project background

Nowadays many children use the bus to go to school. Therefore, due to the large number of students, it is difficult to keep track of students entering and leaving the bus. As a result, accidents may occur when students fall asleep and the bus driver will not be able to notice students leaving, which make them subjected to die from suffocation. This problem has happened before in many countries including Saudi Arabia. According to Arab News (2015), an accident in Jeddah, Saudi Arabia has occurred where an eight years old student has been forgotten in the school bus and left to die from suffocation. Another situation in Saudi Arabia has happened as according to Trad (2018), a young student in Saihat, Saudi Arabia has also died by being left in the bus. Not only that, Trad mentioned that this incident was not the first that happened in that school as similar incident had happened before where the student thankfully had been found before dying. This issue cannot be ignored as it is affecting humans' lives. This problem did not happen once as it happened in other countries as well making it worldwide bus school problem. In addition, it is required in schools to take attendance to keep student records. Teachers often waste time taking attendance. Traditional ways of student's attendance are becoming a waste of time when there are more efficient ways of taking attendance. RFID technology could offer solutions to both problems.

RFID tags have a circuit and an antenna to transfer data to the reader. The project is based on Radio Frequency Identification (RFID) technology as a sensor. The RFID is a wireless communication that use electromagnetic coupling to identify a tag in certain frequency range. The detection comes from the radio waves that the RFID reader produces. RFID technology is a way of commutation that is widely used in industry for various of application. There are two types of RFID: active, and passive. The passive RFID system is composed of tags, and reader the tags operate on different frequency rang to communicate low frequency, high frequency, and ultra-high frequency. The low frequency range between 30 kHz to 300 kHz and can detect the tag up to 10 cm distant. It has disadvantage of low distant reading and with slower data read rate, however it has the advantage of performing better in water and metal. High frequency has a range of frequency 3 MHz to 30 MHz and detect the tags from 10 cm to 1 m distant. It is commonly used in card payments, and data transfer. Ultra-high frequency has a range of frequency 300 MHz and 3 GHz. Its advantage is that it can read tags from high distant that can reach up to 15 m and is cheaper to manufacture. On the other hand, it has disadvantages that it is very sensitive to water and metal. The UHF RFID tags memory mostly consists of tag identifier memory (TID), electric product code (EPC), user memory, and password. Every tag has a special TID code that cannot be changed. In addition, EPC is an edited part of the tag where it can be customized to serve certain purpose. The user memory is an extent of the EPC where if the application required more bytes to identify a certain item further more. Also, the tag contains a password that is used for safety purposes where it could prevent duplication of the ID. The other type is active RFID which sends a single from the antenna to the powered tag that reflect energy back to the reader. The powered tags give the advantage to read for a very high distant.

Previous work offers solutions to both problems using the RFID and GSM technologies. Students can be recognized using RFID unique tags when entering the bus, which makes it easier for the bus driver to keep track of the number of students entering and leaving the bus. Also, SMS massages can be sent as notifications to students' parents to assure them the status of their children. Students can also use their ID tags to mark their attendance and

then the system can automatically save their information in a database. In previous work RFID and GSM technologies are either used in taking attendance or monitoring students riding the bus. This project combines both taking attendance and monitoring students riding the bus, which creates a full school system. Not only that our desired system will have a user interface for the bus driver, and parents to be able to monitor the students and assure the safety of students, but also GSM will be used for notifying the parents of students' movement, and GPS to monitor the speed of the bus and notify the school if the legal speed was exceeded as well as displaying student location while in bus.

2.2 Previous Work

Previous Projects (1):

According to Shabaan, Bekkali, Hanida, &Kadri (2013) in their article, they offer an enhanced safety system for student's transportation using the RFID technology. The project aims to offer a solution for the repeatedly dying or missing students in the period of going or coming from home to school. To do that they used passive RFID inserted in each student bag. The system was designed using RFID, GPS, and GPRS. By having the three technologies (GPRS, RFID, and GPS) they could have some functions that will serve the purpose of safety, and offer detailed information of each student movement to the school and the parents. The block diagram is shown in Figure 2.

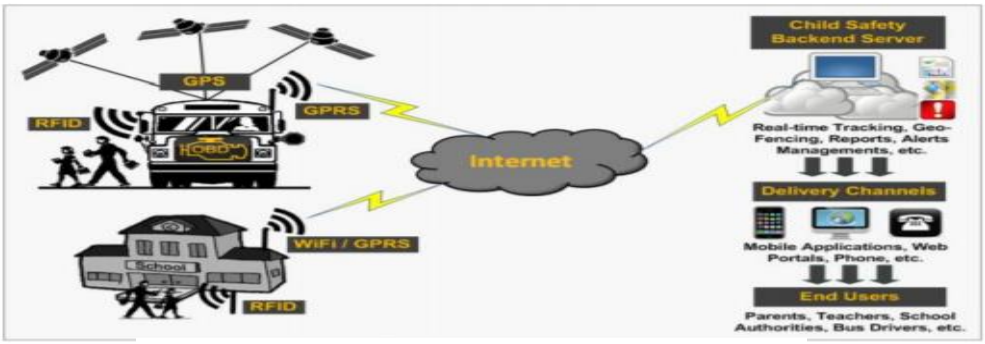


Figure 2: Previous Projects (1) Architecture

Previous Projects (2):

According to ASUNDKAR (2016) project, shows a design for the safety and security of school buses. This project has been designed in a way where it is divided into three units. The first part is the hardware part which is inserted inside the bus. The bus has the RFID tags and reader, GPS module, GSM modem, switch, and microcontroller (Arduino Mega 328). The RFID tag used in this project is passive, and the reader is placed in the entering door of the bus. The passive RFID tag is within the student's ID card. The GSM modem purpose is for communication. the GPS has a feature of tracking students at the bus and send SMS if the bus if the bus driver exceeded the speed limit. There also another unit which is given to the parents. In this unit parent can have an android application on their phone to get the necessary information about their child. Third unit is at the school. This unit plays as the controlling unit for adjust any information and keeping record of students on every move. The block diagram is shown in Figure 3.

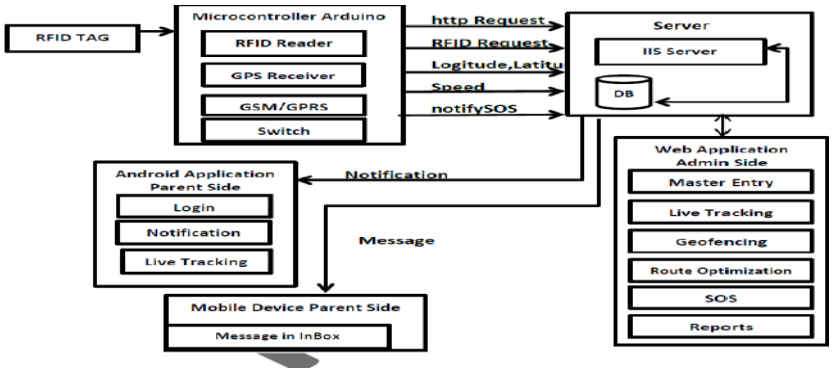


Figure 3: Previous Projects (2) Architecture

Previous Projects (3):

According to AlLawati, AlJahdami, AlBlushi, Aladawi, & Aladri (2016), in their paper they explain how to use RFID tags in order to assure the safety of students. The designed system focuses on two parts: one on the bus and the other at the school. In the bus, there is a reader that gets the information from the tag when it is closed by then send it to the database at the school. The database at the school gets the information from all the buses and checks who is in and who is not then takes action if needed. The action is to send SMS to the student parents. There is also a web-based application. The block diagram is shown in Figure 4.

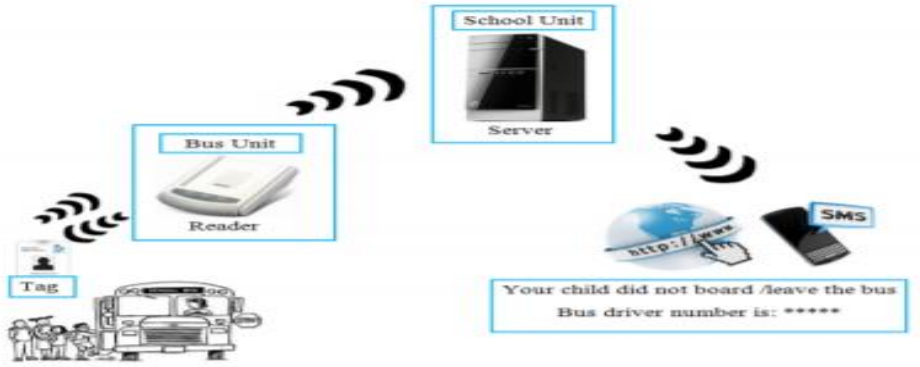


Figure 4: Previous Projects (3) Architecture

Previous Projects (4):

According to Wehab, Yusof, Kadir, Sanudin & Tomari (2009), in their article they claim to have made an attendance recording using active RFID. They compared other technologies as well as active RFID. They state that the best for the project is to use active RFID. After deciding the best technology, they had to include the important tools. They used visual basic 6.0 to design the user interface. For the database they have used Microsoft Access. Also, one tool is the RFID tag, and reader. To summarize, they created a diagram in which you can see the development module which is shown in Figure 5.

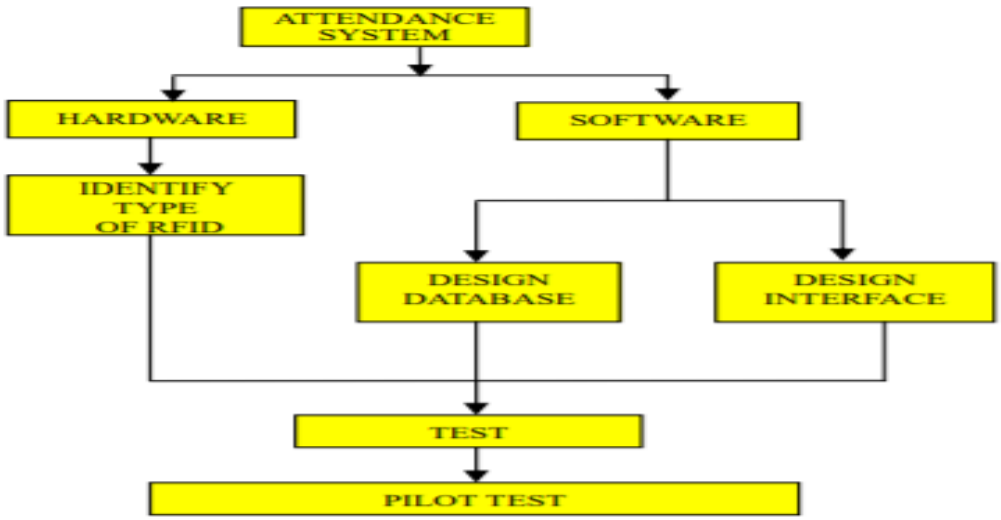


Figure 5: Previous Projects (4) Architecture

2.3 Comparative Study

Here is a compressing of our project and previous based on the criteria of GSM, type of RFID, attendance system, bus tracking, and database as shown in Table 1.

Table 1: Comparitive table

Projects	1	2	3	4	Our Project
GSM	GPRS	√	√	√	√
RFID	Passive	Passive	UHF Passive	Passive	UHF Passive
Attendance				√	√
Bus track	√	√	√		√
Student record	√	√	√	√	√

3. System Design

3.1 Design Constraints

*3.1.1 **Design Constraints:*** System consideration

In the design, the aim of the project is to provide a solution with a consideration of one class room with 10 students only. Another consideration is that all students of the class are going and leaving school via the same bus.

*3.1.2 **Design Constraints:*** Bus user interface

In the design of the project, the bus system requires the bus driver to be having a smart phone with either IOS or android. The phone must be charged in order to be able to control the system otherwise the system will not work as intended.

*3.1.3 **Design Constraints:*** Lost card

The design of the project assumes that all student will have their ID all of the time. If a student lost their ID, it would be a problem as there is no manual solution. The student will be considered absent and have not ridden the bus making him outside of the designed system.

3.2 Design Methodology

The project can be divided into two units: one at the school and the other at the bus sharing the same database. The methodology that the project is aiming to do is RFID based for detection of students, with GSM for alarming messages, GPS for speed measurement and live location of the bus, database to store and call student information, and controller to be able to control the systems.

3.3 Product Sub-systems and Components

3.3.1 Product Sub-system 1: RFID reader (Attendance)

Communication: Radio frequency communication between the reader and the tags.
Control: Arduino Mega.
Power: Reader is powered by 220 V or 9 V battery.
Alternative options: Passive RFID, or active RFID.

3.3.2 Product Sub-system 2: RFID reader (Bus)

Communication: Radio frequency communication between the reader and the tags.
Control: Arduino Due.
Power: Reader is powered by Arduino mega.
Alternative options: Passive RFID, or active RFID.

3.3.3 Product Sub-system 3: GSM

Communication: Communicate with the user through cellular network.
Control: Micro-controller to program the GSM.
Power: Powered by a battery.
Alternative options: GPRS, or Wi-Fi.

3.3.4 Product Sub-system 4: GPS

Communication: Global Navigation Satellite System network.
Control: Compiler to program the GPS.
Power: Powered by a battery.
Alternative options: No alternative.

3.3.5 Product Sub-system 5: Graphical user interface using RemoteXY

Communication: Wi-Fi access point.
Control: ESP 32.
Power: Charged phone.
Alternative options: Blynk, Druid Builder.

3.3.6 Product Sub-system 6: Graphical user interface using Blynk

Communication: Wi-Fi.
Control: NodeMCU.

Power: Charged phone.
Alternative options: Druid Builder.

*3.3.7 **Product Sub-system 7:*** Database

Communication: The internet.
Control: School computer & Arduino.
Power: School computer.
Alternative options: Microsoft Access.

3.4 Implementation

The design of the system will be controlled by Arduino. The Arduino will control UHF RFID readers where the tags can be detected of a range of at least 5 meters. There will be GSM connected to the Arduino to send SMS. Also, all of the readings will be stored in the database where a student record is created. In addition, the programming logic will allow detection of students not leaving the bus and will create an alarm. These operations will assure meeting the targeted specifications. Since the project is mostly programming based on Arduino, most of the testing was tested on the serial monitor then applying it to see the response. Each system was created in a code by itself then integrated with the main code having every step tested.

In the project, there were alternative solutions for the RFID, SMS, user interface, and database. The first consideration was active RFID reader instead of UHF RFID. Due to the poor documentation of active RFID in the market and the price of the tags, it was better to go with the UHF reader. For the SMS it was considered to pick GPRS, Wi-Fi, or GSM. GSM was chosen as it was the best when it comes to simplicity as well as reliability. For the user interface there were wide options of usage whether it was website, touch screen, or phone application. Since it is the 21st century where almost every person has a smart phone, it was best to use it for user interface. The user interface for bus driver requires a switch to pick the modes as well as students' name making RemoteXY the perfect platform to interface. On the other hand, parents' user interface required more features such as tracking map. To meet the parent's requirement, Blynk was the best option as it can be connected to Google Maps easily and have the location displayed there. Also, Blynk can offer all the other requirement of parents' user interface. When it comes to database, there are many different ways of creating it, but MySQL is the most popular, reliable, and flexible way of having database.

4. System Testing and Analysis

4.1 Sub-system 1: Bus

This is the first main sub-system that is used to count the number of students riding and leaving the bus using RFID technology and record the IDs in the database as well as assuring that there is no student left on the bus. In addition, the sub-system will use GSM to send SMS messages and a GPS to update the bus location and find the speed. Under this sub-system, there are four sub-sub-systems which are the following:

Sub-sub-system#1: A **UHF RFID reader** is programmed to read tags with access control to registered tags. The reader is the main component of the system and it is capable of reading with a range up to 5 m distant with operating frequency of 865 MHz. Testing the UHF reader with the bus coding logic was independent of other sub-systems. In the desired system it should be switching modes as a command from the bus user interface. In testing the system it was instead connected to a switch that could switch the modes to start, and arrival. The setup of the system is shown in Figure 6. The ouput response is tested in a serial monitor by showing the recognition of the student, the mode of operation, the flag status number that keep the students ID to be read only once in each mode, and the numbers of students absent, attended, and arrived. We first start the testing by creating a senario of students first entering then at the end leaving the bus. First, we start by putting the mode in starting mode. Then, we swipe the first card that correspond to the first student (Saad). In Figure 7 the out response is printed showing that Saad was recognized then increaseing the number of students attended, decreasing the number of absent students as well as setting the reading flag to be one . Then, in Figure 8 another student (Ali) enters the bus. Next, the mode is switched to arrival mode. Arrival mode resets the reading flags to allow students ID to be read again and count the number of ariving students. In Figure 9 first student (Ali) arrives at the school increasing the number of arriving students. In addition, since the number of arriving students is less than the students who entered the bus a warning is created. After that, another student (Saad) arrives removing the warning message as all students who entered the bus have arrived as shown in Figure 10. By observing these responses we assure the working of the reading logic.

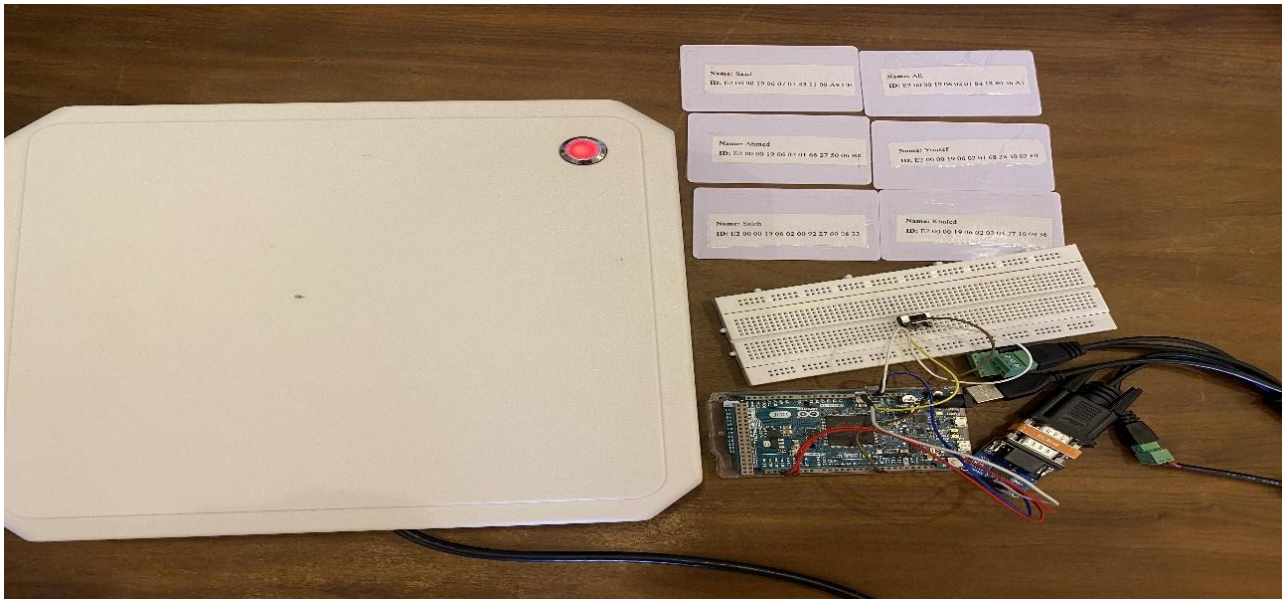


Figure 6: UHF reader testing set-up

```
checking id
110A9CEid matched Saad
Attended student & absent Saad +1 -1
Saad here
Arrival: 0Start: 1
0 1 0 0 0 0 0 0 0 0
No. of absent:
9
No. of attended:
1
No. of arrived:
0
Out
```

Figure 7: Serial monitor result

```
checking id
188056Alid matched Ali
Attended student & absent Ali +1 -1
Ali here
Arrival: 0Start: 1
0 1 1 0 0 0 0 0 0 0
No. of absent:
8
No. of attended:
2
No. of arrived:
0
Out
```

Figure 8: Serial monitor result

```
checking id
188056Alid matched Ali
Arivved student Ali +1
Ali here
Arrival: 1Start: 0
0 0 1 0 0 0 0 0 0 0
No. of absent:
8
No. of attended:
2
No. of arrived:
1
Warning not all students have left the bus
Out
```

Figure 9: Serial monitor result

```
checking id
110A9CEid matched Saad
Arrived student Saad +1
Saad here
Arrival: 1Start: 0
0 1 1 0 0 0 0 0 0 0
No. of absent:
8
No. of attended:
2
No. of arrived:
2
Out
```

Figure 10: Serial monitor result

Sub-sub-system#2: <u>User Interface</u> is designed using RemoteXY which is a mobile based user interface. RemoteXY allows the bus driver to pick the trip mode as well as display in bus, absent, and arrived students' names. The system intended to be connected with Arduino to receive the information of student's names, the number of attended, absent, and arrived students via serial communication. In the future, it should be connected with the main Arduino with serial pins to receive information and two digital pins connected to Arduino to indicate the modes of operation. To test the sub-system independently, an ESP32 shown in Figure 18 is programmed directly with Arduino IDE with nothing connected to it. After that, using Wi-Fi access point it can be connected to a phone whether Android or IOS. The system has two modes start mode and arrival mode. To test the working of the modes without Arduino two virtual LEDs are programmed to be on when the corresponding mode is chosen. In addition, text indication in the user interface for trip mode will indicate the mode as shown in Figure 11-12. When two mode are picked an error will occur informing the bus driver that one mode should

be chosen at a time as shown in Figure 13. To test the user interface furthermore, the ESP32 was fed from Arduino IDE of a scenario that Arduino could sends. The ESP32 was fed with student name (Saad), number of absent (9), attended (1), and arrived (1) students. In this case we assumed that the student who entered the bus has arrived. From these information RemoteXY can display the names of the students in the right place as shown in Figures 14-16. Also, it assures that there is no empty space between two students in display. In other words, Figure 15 if the first student attended, his name will disappear from the absence list and replace with the next student.

Figure 11: RemoteXY interface

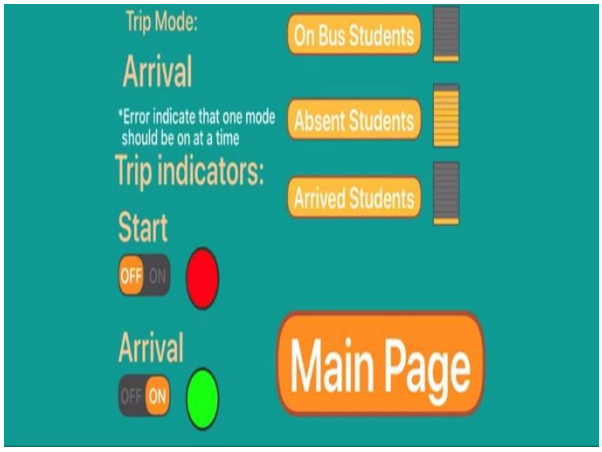


Figure 12: RemoteXY interface

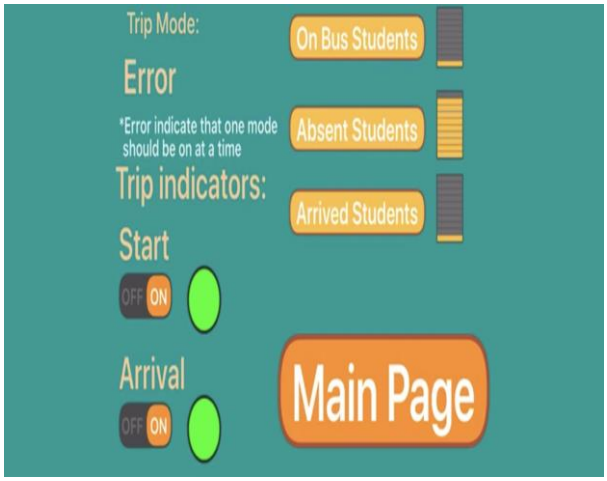


Figure 13: RemoteXY interface

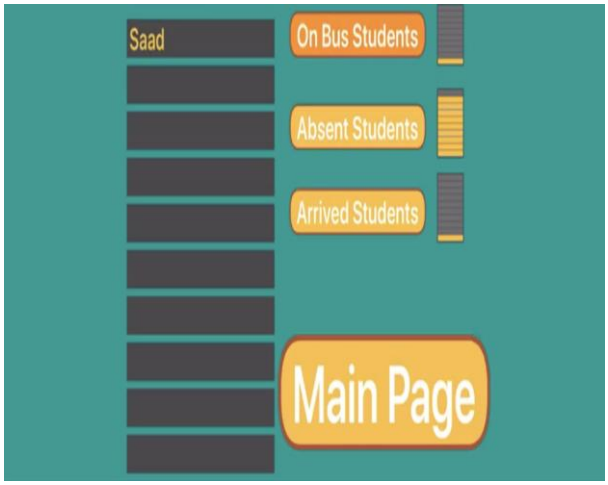


Figure 14: RemoteXY interface

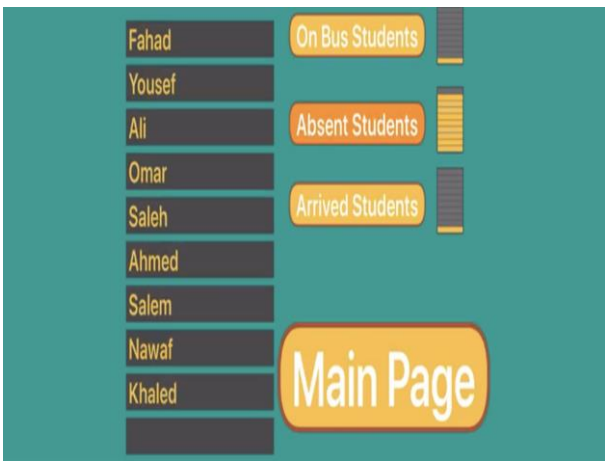


Figure 15: RemoteXY interface

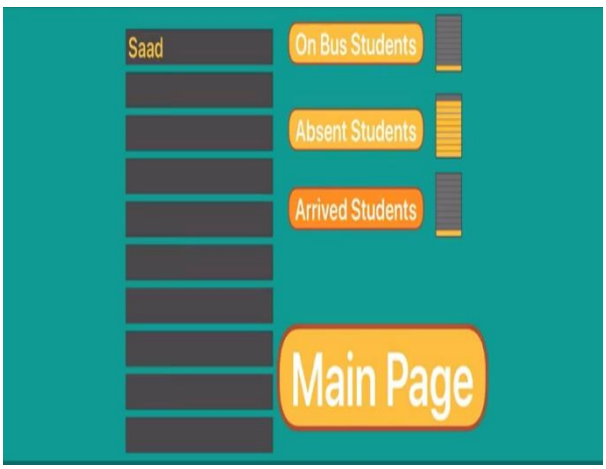


Figure 16: RemoteXY interface

Sub-sub-system#3: GSM is used to send SMS messages to parents when student ride the bus and arrive at the school. Since the GSM is connected to Arduino Due, it has to operate at 3.3 V. Unfortunately, most GSM shields operate at 5 V which might damage the Arduino Due if used. Therefore, a GSM module with independent power source is used as shown in Figure 17. Then, the GSM should communicate with the Arduino through serial port. The coding and use of GSM technology have been experienced with the classroom attendance system. Therefore, the only part left is making the module communicate with the Arduino serially.

Sub-sub-system#4: GPS is used to determine the location of the bus and measure the speed. A GPS module shown in Figure 19 is connected to NodeMCU shown in Figure 20 to update the location, and speed for the parent's user interface. In addition, if speed limit has been exceeded the NodeMCU will send the information to the Arduino to command the GSM to send an SMS to the school.

Figure 17: GSM module

Figure 18: ESP32

Figure 19: GPS module

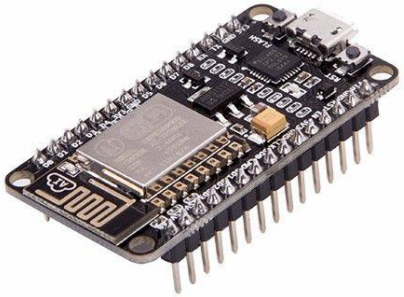


Figure 20: NodeMCU

4.2 Sub-system 2: Class

The class is the second main sub-system and it is designed to have initially passive RFID reader to recognize student IDs. In the future we plan to upgrade the system to UHF passive. We Already acquired the UHF reader to upgrade the system but we could not do so due to the COVID-19 challenge. The system operates by reading the IDs then it sends the data to the database with exact date and time. Moreover, SMS messages will be sent to the parents to notify their children attendance status. The system is designed to read the IDs of the students who are registered in the system and ignore the rest. Also, the system is capable of determining the number of students attending the class, being late, or being absents. Under this sub-system there are two Sub-sub-systems.

Sub-sub-system#1: This sub-system is based on **passive RFID** using RC255 module shown in Figure 22. This module operates at a frequency of 13.56 MHz and designed to be used with Arduino. Therefore, it is easier to start an RFID based project with a simple reader. Hence, the project started with this RFID module to create attendance system. To test the work of the RFID, the setup is shown in Figure 21. After implementation and coding the passive RFID will be ready for students to scan their tags with the LCD displaying "Please swipe your card" as shown in Figure 23. Each tag has student's name as shown in Figure 24. After the student swipe his card the LCD will welcome the student by his name as shown in Figure 25 and it indicate that he is attended as shown in Figure 26. The system will also indicate if the student is on time or if he is late as shown in Figure 27. The buzzer will beep when students swipe their cards as shown in Figure 28. The system also has access control when a student who is not registered to the class swipe his card the LCD will display "wrong class" and the name will not be recorded in the database as shown in Figure 29. Initially to test students' attendance record an SD card is connected to the system to record data in a text file as shown in Figure 30.

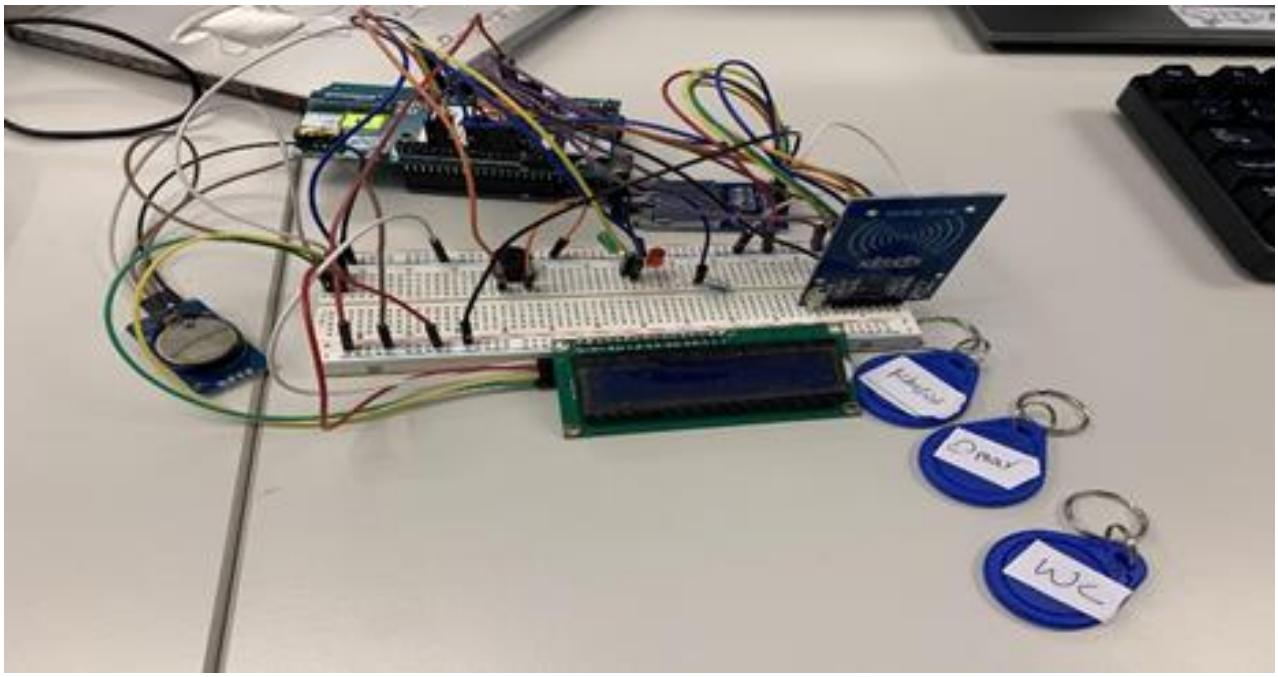


Figure 21: Attendance system

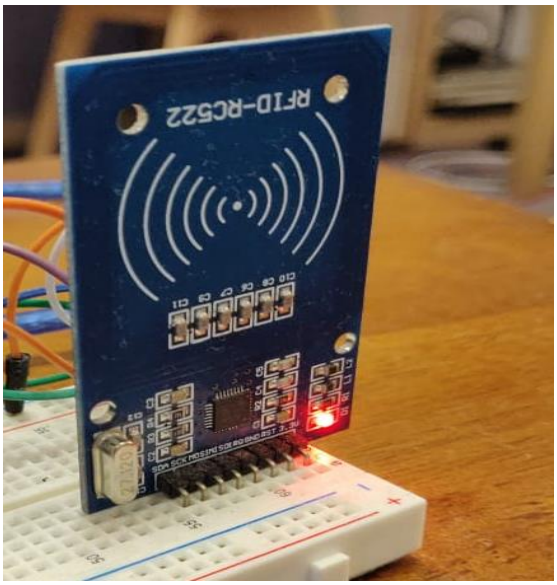


Figure 22: passive RFID reader

Figure 23: LCD display

Figure 24: RFID tags

Figure 25: LCD display

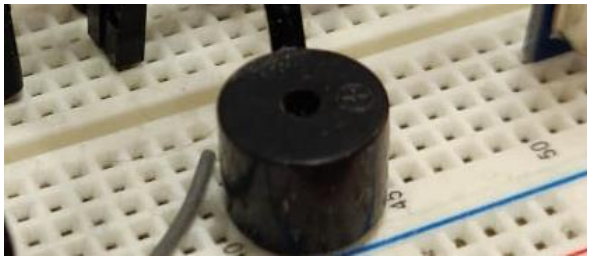


Figure 28: Buzzer

Figure 27: LCD display

DATA - Notepad

File Edit Format View Help

```
Khaled , 2020/2/2,22:22
Khaled , 2020/2/2,22:23
Khaled , 2020/2/2,22:22
Fahad , 2020/2/3,22:36
Khaled , 2020/2/3,22:36
Omar , 2020/2/3,22:36
Omar , 2020/3/3,9:18
Ahmed , 2020/3/3,9:23
```

Figure 30: Student records

Figure 29: LCD display

Sub-sub-system#2: This sub-system will be using **GSM** technology to send SMS messages. It will use a GSM shield with integrated antenna as shown in Figure 31. The GSM will be responsible of sending SMS to parents when a student attend with exact date and time as well as indicate if the student is absent. To test the working, it will go through a scenario where a student swipe his card and the LCD will display "message sent" as shown in Figure 32. Then, the student will be marked attended as shown in Figure 33. Next, an SMS message will be sent to the student's parent with exact date and time as shown in Figure 34. When the class starts an SMS message will be sent to the absent student's parent as shown in Figure 35.

Figure 31: GSM shield

Figure 32: LCD display

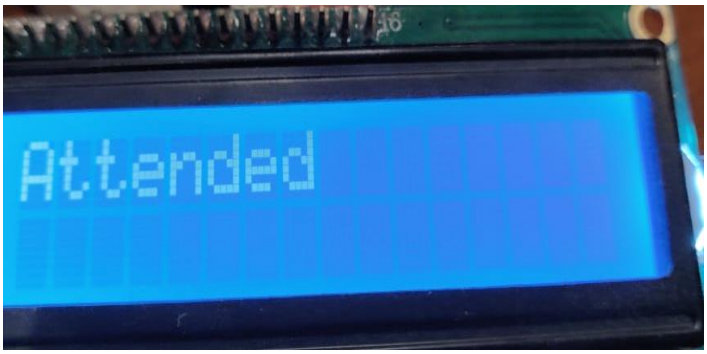


Figure 33: LCD display

Figure 34: SMS attendance message

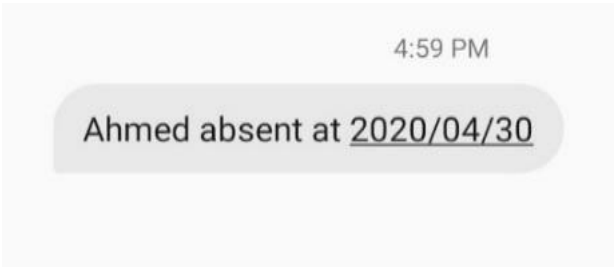


Figure 35: SMS absence message

4.3 Sub-system 3: Database

Using MySQL, a **database** is created at local server that can be accessed from the school administration computer. The database is setup in way that have options of adding new students' ID and monitor student attendance Figure 36 show the links that provide the database features. In the database the administrator can register new students' names, IDs, and gender as shown in Figure 37. All students' information will be displayed with IDs, date, and time with indication if they were on time or late as shown in Figure 38. The future plan for the database to exchange information with both sub-systems bus, and class via Wi-Fi. In other words, the bus and the class sub-systems will be responsible of updating the database of students' status, where the database will be storing the information to be visible for the school. In addition, the database will be connecting the whole systems information into one storage unit. Therefore, the database is the overlap that connect the two main subsystems bus, and class.

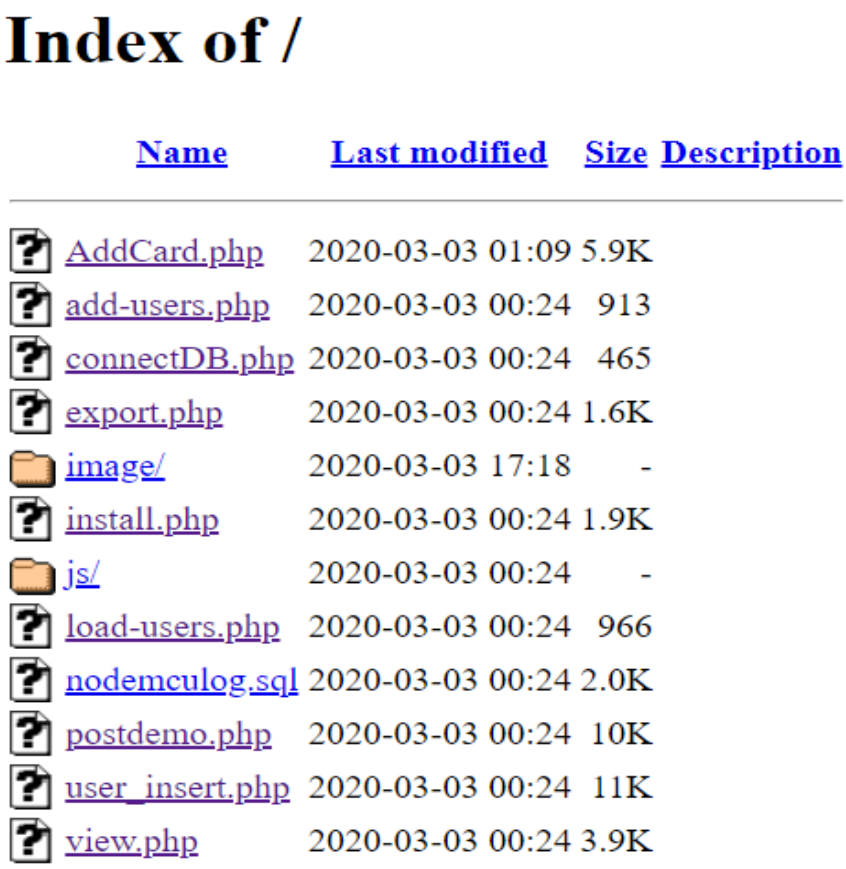


Figure 36: Database links to all options

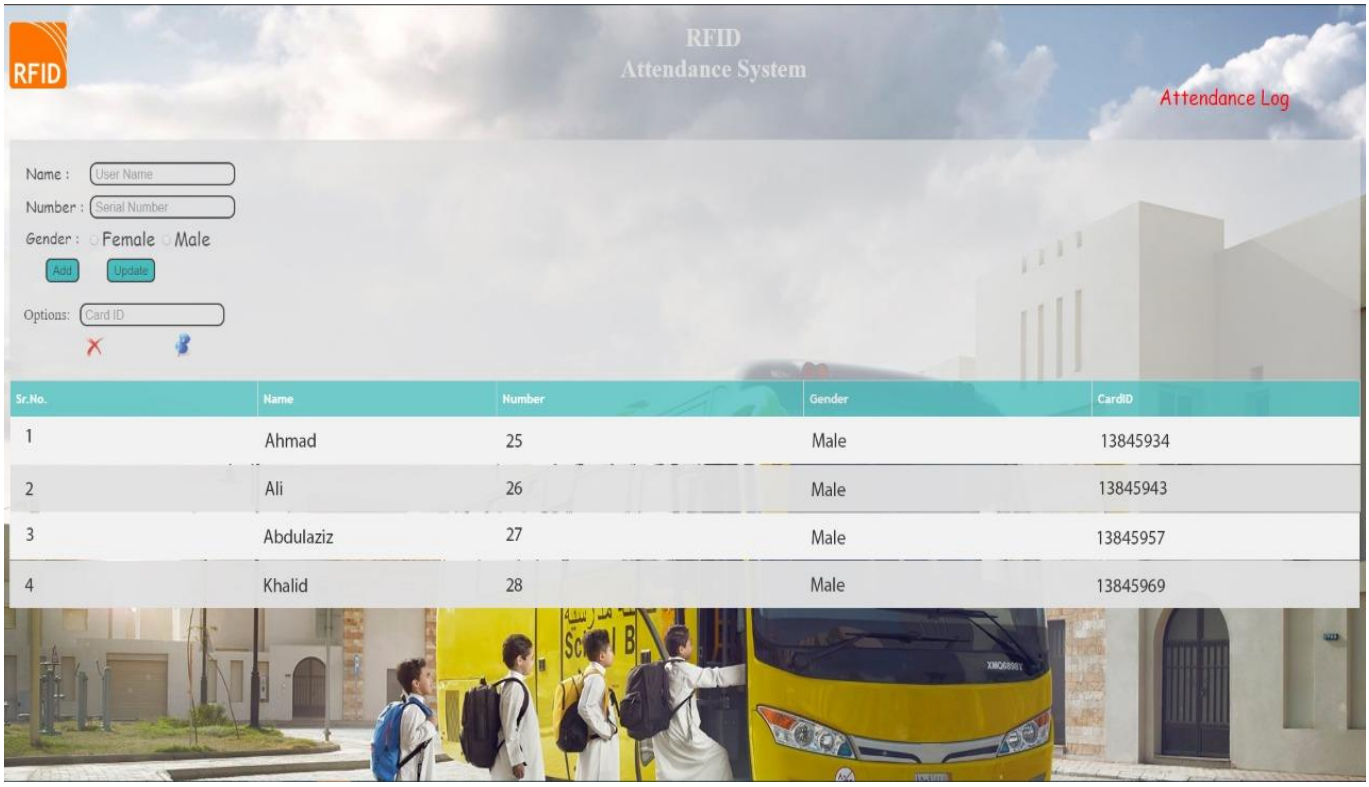


Figure 37: Registering page

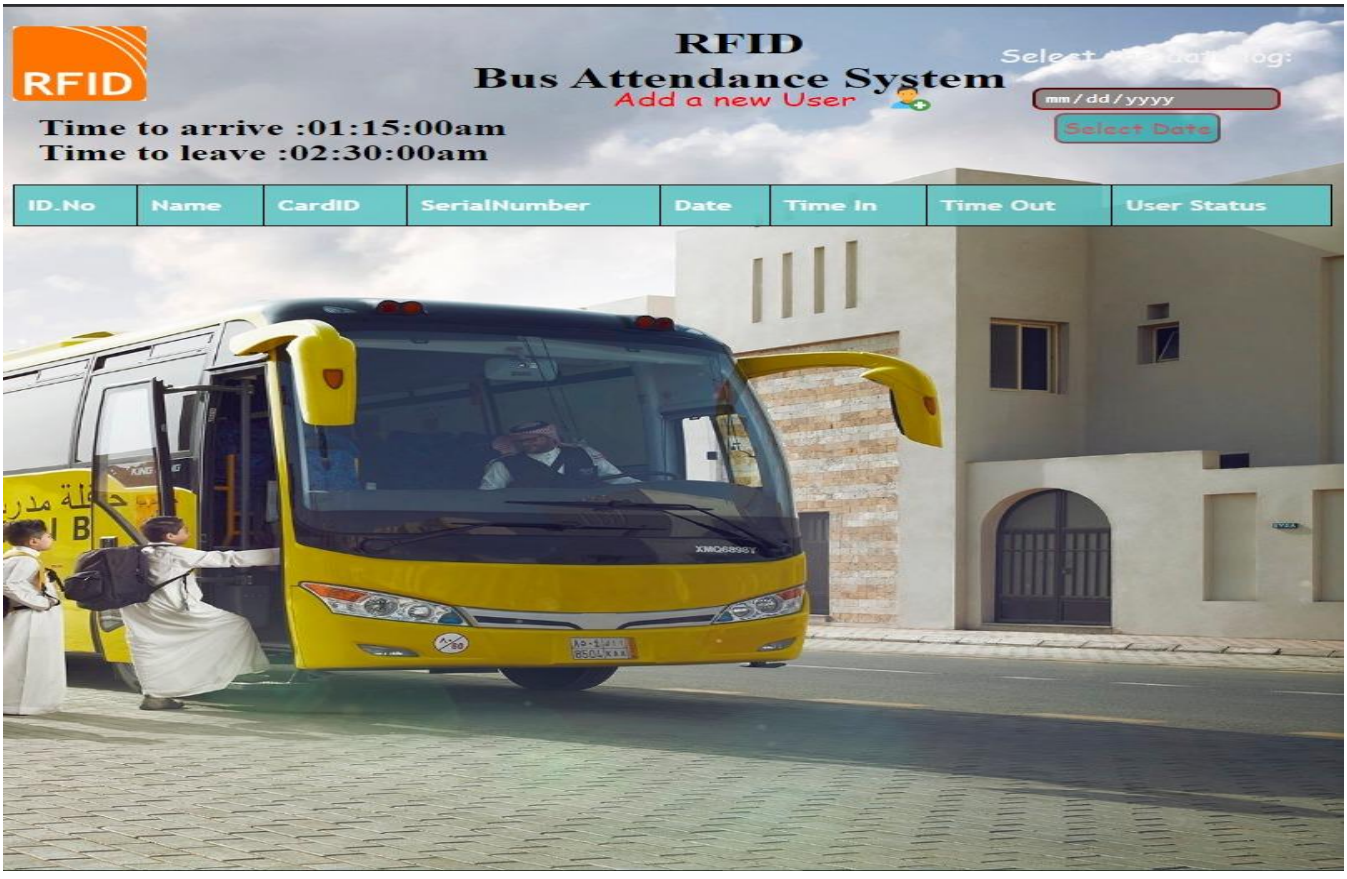


Figure 38: Information display

4.4 Sub-system 4: Parents user interface

This **user interface** is designed for the parents using Blynk which is a platform that allows the user to build infrastructure to control the Arduino. The graphical user interface allows parents to have different features to ensure the safety of their children. The user interface is programed with NodeMCU via Arduino IDE. The first feature is that the system gives the parents the option of tracking the bus using Google Maps. Second, there is a speedometer where the speed will be displayed for the parents on virtual LCD. In addition, if the bus has exceeded the legal speed, NodeMCU will notify the Arduino to send an SMS message to the school indicating that the bus driver is speeding. Third, in future development parents will be able to enter their child ID number to get access for the number of attendances and absences. The application will communicate using Wi-Fi with the database to receive attendance information about students. Figure 39 shows the parents user interface.

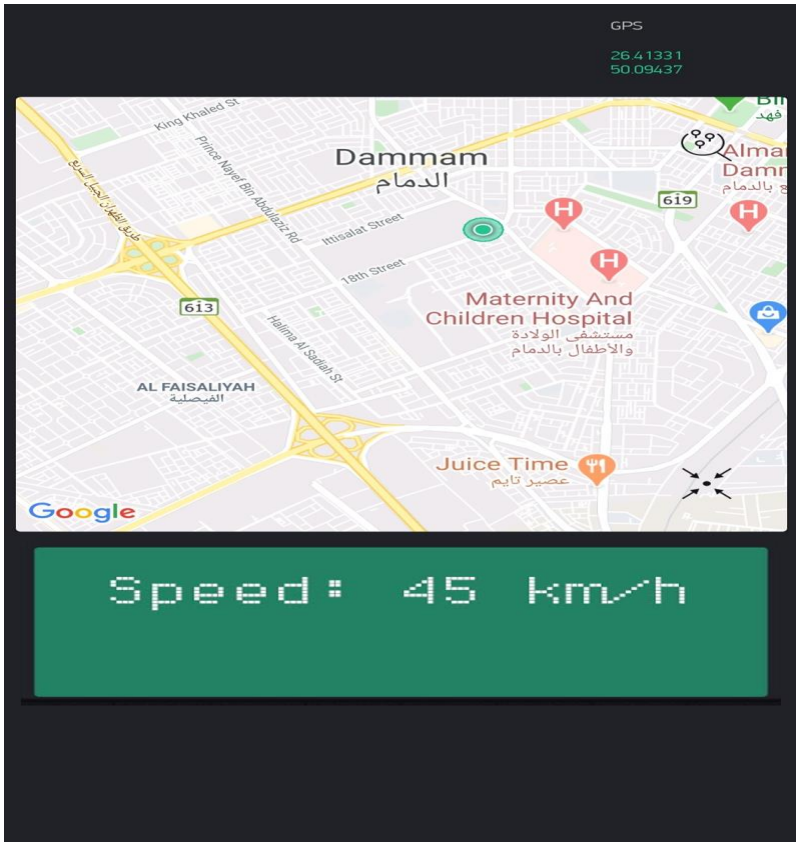


Figure 39: Parents user interface

4.5 Overall Results, Analysis and Discussion

In this project we dealt with devices that were unfamiliar to us, so we did a lot of testing to understand how it works. These devices are:

- **Active RFID reader**: the active RFID reader is programmed using C++ visual studio; however, we were not familiar with this programming language. We thought it would be easier to program it using Arduino, so we connected the interface port from the

reader to USB level using FT232RL as shown the Figure 39. We tried to send command to reader to read the ID but we did not receive and response. That is why we stopped using the active reader. In addition, the active RFID reader was poorly documented. The command for the active reader exists from the manual as shown in Table 2.

Table 2: Acitve reader command sample

NUM	COMMAND	CODE	COMMENT
1	CMD_READ_SYSTEM_PARAM	0x10	Read system param. Eg: Send:53 57 00 03 FF 10 44 Recv:43 54 00 0D 00 10 01 14 11 C3 DD 93 8E 17 01 23 2A 14: SoftVersion 1.4 11:HarVersion 1.1 C3 DD 93 8E 17 01 23:DevSN
2	CMD_READ_DEVICE_PARAM	0x20	Read device param Eg: Send:53 57 00 03 FF 20 34 Recv:43 54 00 21 00 20 01 C3 55 01 00 00 00 0A.... C3:DevType (C3 Means 5300) 55:Default param switch. None 55 default param. 01 00.....:Params,See CMD_SET_DEVICE_PARAM Command

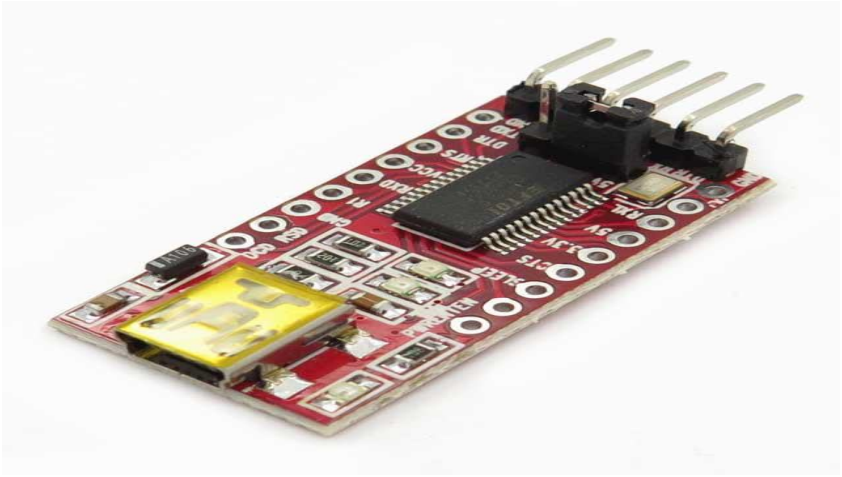


Figure 39: FT232RL

- **UHF RFID reader:** the same case for UHF reader, we wanted to ingrate it with Arduino to ease the programing process. We used RS232 to TTL level as shown in Figure 41. We sent the ID reading commands stored in arrays from the Arduino Mega to the reader to read the tags as shown in Figure 40. The received ID information was not always accurate. Therefore, the reader required higher processing speed making Arduino Due a better option. Then, we were able to get a perfect response on initialized command as shown in Figures 42. where the ID separated by stars is the full ID and the one separated with dashes is the extracted ID.

```
const char RdTagAM2[] PROGMEM ={0x53,0x57,0x00,0x0A,0xFF,0x02,0x01,0x02,0x06,0x00,0x00,0x00,0x00,0x42};
const char IDcmd[]    PROGMEM ={0x53,0x57,0x00,0x03,0xFF,0x20,0x34};
```

Figure 40: Send command to the reader

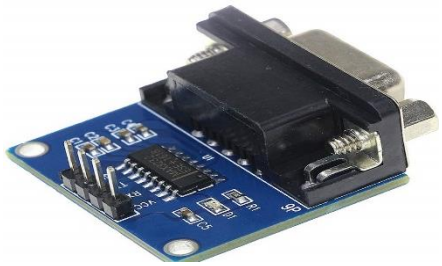


Figure 41: Rs232

43*54*0*10*0*2*1*E2*0*0*19*6*2*2*4*27*10*9*56*B7*20

read id

E2-0-0-19-6-2-2-4-27-10-9-56-for loop

Figure 42: the response to the command with ID number

- **Wi-Fi communication:** We needed the Wi-Fi communication to create interface that is accessible by the phone. We first started testing using EPS8266 and the communication was not done properly. Then, we tried using NodeMCU and ESP32 which worked perfectly. We performed test codes and tried almost all the functions that we need for our project.

- **Real time clock (RTC):** When working with Arduino Due we can only connect 3.3 V components, while the RTC is 5 V. After searching, we found out that Arduino Due has RTC built in the board.

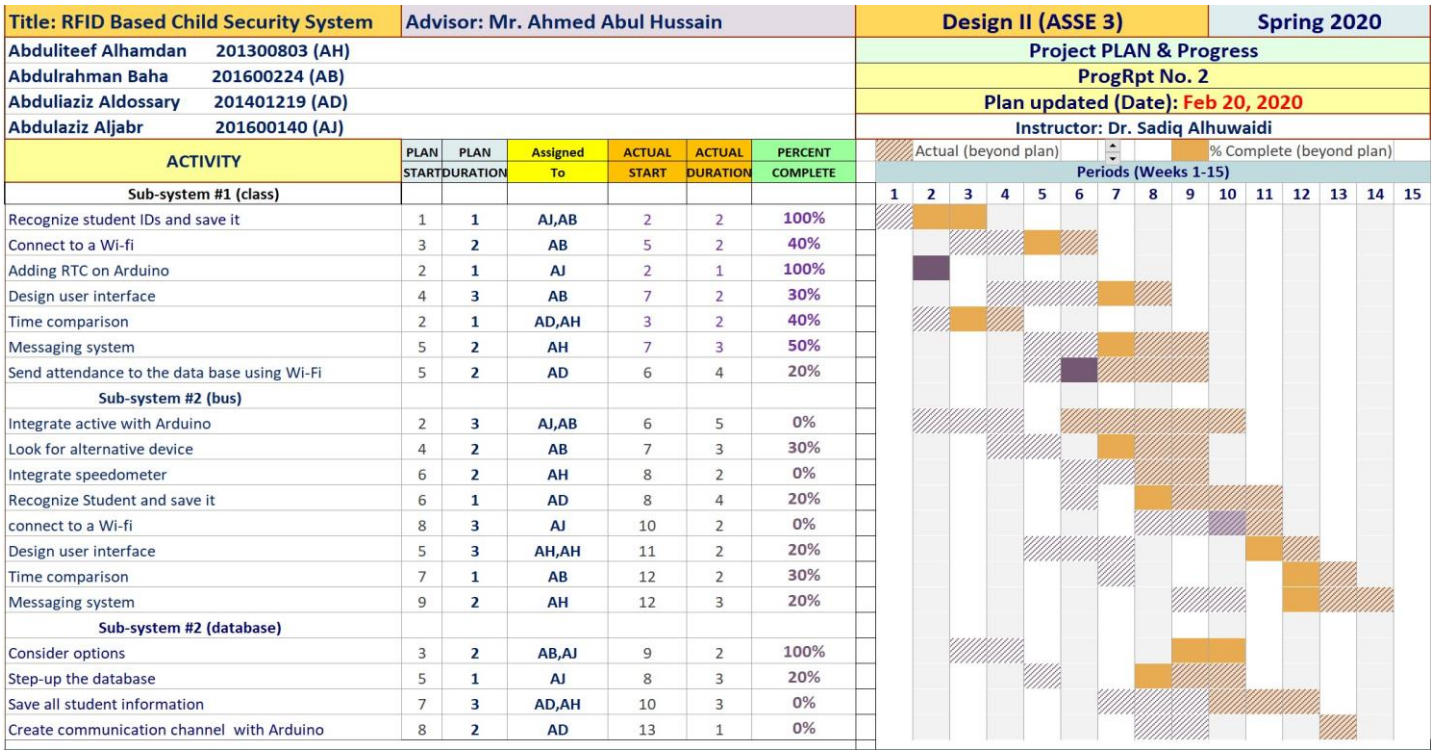
- **SIM808 GPS/GPRS/GSM Shield**: This Arduino shield comes with a library and we tested all its functionality needed for the project. We tested the GPS to find the location, and the GSM to send a message. The shield is shown in Figure 43.

Figure 43: SIM808 GPS/GPRS/GSM Shield

5. Project Management

5.1 Project Plan

After dividing the work to three main sub-systems, we made a plan from now until the end of the semester with specific tasks and specific duration. Each group member is assigned with a certain task to do as shown Figure 44.



Title: RFID Based Child Security System		Advisor: Mr. Ahmed Abul Hussain				Design II (ASSE 3)	Spring 2020
Abduliteef Alhamdan	201300803 (AH)					Project PLAN & Progress	
Abdulrahman Baha	201600224 (AB)					ProgRpt No. 2	
Abduliaziz Aldossary	201401219 (AD)					Plan updated (Date): Feb 20, 2020	
Abdulaziz Aljabr	201600140 (AJ)					Instructor: Dr. Sadiq Alhuwaidi	

ACTIVITY	PLAN START	PLAN DURATION	Assigned To	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Sub-system #1 (class)						
Recognize student IDs and save it	1	1	AJ,AB	2	2	100%
Connect to a Wi-fi	3	2	AB	5	2	40%
Adding RTC on Arduino	2	1	AJ	2	1	100%
Design user interface	4	3	AB	7	2	30%
Time comparison	2	1	AD,AH	3	2	40%
Messaging system	5	2	AH	7	3	50%
Send attendance to the data base using Wi-Fi	5	2	AD	6	4	20%
Sub-system #2 (bus)						
Integrate active with Arduino	2	3	AJ,AB	6	5	0%
Look for alternative device	4	2	AB	7	3	30%
Integrate speedometer	6	2	AH	8	2	0%
Recognize Student and save it	6	1	AD	8	4	20%
connect to a Wi-fi	8	3	AJ	10	2	0%
Design user interface	5	3	AH,AH	11	2	20%
Time comparison	7	1	AB	12	2	30%
Messaging system	9	2	AH	12	3	20%
Sub-system #2 (database)						
Consider options	3	2	AB,AJ	9	2	100%
Step-up the database	5	1	AJ	8	3	20%
Save all student information	7	3	AD,AH	10	3	0%
Create communication channel with Arduino	8	2	AD	13	1	0%

Figure 44: Project plan

5.2 Contribution of Team Members

Table 3 states the most major tasks of the project and a percentage of each team member contribution in that task.

Table 3: Controbution of team memerbs

Task	A.AlDossary	A.AlJaber	Abdulittef	Abdulrahman	Task Total
Pregaming the reader	**15%**	**40%**	**15%**	**30%**	**100%**
Wi-Fi communication	**10%**	**40%**	**10%**	**40%**	**100%**
Interface	**15%**	**35%**	**15%**	**35%**	**100%**
Database	**30%**	**20%**	**30%**	**20%**	**100%**

5.3 Project Execution Monitoring

- **Meetings with Advisors:** Every weak we have a group meeting with our advisor to discuss the progress of the project and to see each team member contribution. We also discuss issues that we faced and try to solve them together. We always keep in touch with our advisor to keep him updated. Many of the issues that we faced were solved by getting consultation from the advisor.

- **Team meetings:** Our team meet weekly outside of the university to work on project. There is work that is done with a group and other work is done individually. In each meeting we try to see each member's progress, discuss new ideas, and solve problems.

- **Buying the components:** When we need to buy new component, we search about it and discuss our options. Then, when we want to buy something, we consult the advisor and get the team approval.

5.4 Challenges and Decision Making

We have been working on the project from almost two semesters and during these times we faced many challenges and difficulties:

1- **Integrating the UHF reader with Arduino:** The reader we bought was not designed specifically to be used for Arduino. It has different communication interfaces including RS232. We had to try to understand the communication interface. We try to communicate serially and send commands to the reader to get response and it did not work immediately. We tried every command, and at the end it worked when we had to set it up from an operation that was not mentioned for it being the set up. It had to be sent first then send the other command for reading. After it worked, we had to make sure to take all of capabilities of the reader and make it as fast as we could. We had to change the Arduino from Arduino Mega to Arduino Due as it is five-times faster processing. With Arduino Due it almost worked perfectly. We had another problem of data being stored in different places with unreasonable values. It was solved by using avr/pgmspace library where it worked perfectly. After doing this we successfully were able to make it work.

2- **Wi-Fi:** We bought a component ESP 8266 to communicate the mobile user interface with the Arduino using Wi-Fi. We tried working with ESP 8266 but it did not serve our purpose. Therefore, it was better to switch to buy another component NodeMCU & ESP32 which was faster and did the job.

3- **Database:** The database was something that we had to learn to use since it is new to us. We read a lot about databases and we saw previous project until we decided to work with MySQL.

4- **User Interface:** The interface is also a new thing to us that we had to learn. The main problem was in coding and how to meet the desired output.

5- **COVID-19:** Due to the coronavirus all schools were suspended and there was a curfew, so it limited our ability meet and do the work. This was a huge barrier that stopped the progress of our project. The components were distributed between group members making it impossible to proceed while we are in quarantine.

5.5 Project Bill of Materials and Budget

At the beginning of the project, we estimated a budget to our project which was 3500 SR and we thankfully we did not exceed that budget. Table 4 shows the whole expenditure for the project.

Table.4: Budget

No.	Description	Quantity	Unit Cost (SR)	Total Cost (SR)
1	RFID tags	20	8	160
2	Reader	3	650	1950
4	GSM	1	180	180
5	Microcontroller	2	120	240
6	Others	NA	150	150
Totals				2,680

6. Project Analysis

6.1 Life-long Learning

Working on the project, there was a learning curve where we had to learn new skills dealing with new technologies to us. Mostly it was self-learning based on information from the internet.

- **Programing skills:** As electrical engineer, we have a good background in programing. However, the project required more advanced programing skills that we needed to obtain.
- **Learning about RFID technology:** To buy the reader, we were looking and learning about types of RFID and how it works. We communicated with many venders asking about the product features and reading datasheets trying to find something that will meet the project specifications.
- **MySQL:** To set-up the database, we had to learn how to use MySQL and understand its features and adapt it to our project. We also had to learn how to communication data between the Arduino and the database.
- **RemoteXY:** To design the bus user interface**,** we used a graphical design application where we had to learn how to use it and adapt the code to our project.
- **Blynk:** Another graphical design interface application that has different features than RemoteXY. We had to learn how to use Blynk to find the location in Google Maps and also print the speed on LCD.

- **Team work:** Working with a team for a whole year to achieve specific goals was quite challenging. We faced many problems and together we tried to find solutions. We were able to enhance our communication skills to understand each other to aim to a common goal.

6.2 Impact of Engineering Solutions

This project provides a solution to a common problem for elementary schools which aims to ease the process taking attendance and ensure students' safety. The project provides autonomous counting of students entering and leaving the bus with high accuracy to ensure no students are left on the bus to die from suffocation. Also, autonomous attendance with SMS message sent to the parents to save time rather than manual attendance. The bus driver can easily check that everyone left the bus with his phone. Parents can see the exact location of the bus, receive SMS notification of students' attendance status, and also see the speed of the bus. This way parents do not have to worry about their children going to school.

6.3 Contemporary Issue Addresses

Due to the large number of students in elementary schools, it is hard for the school's administration to keep track of all students. Students' safety is a huge responsibility, starting from riding the bus until they are delivered to their destination. Using manual methods to keep track of students will consume time and effort and it is not very accurate. In the past many tragic incidents happened in Sadia Arabia where student slept in the bus and the driver lock the bus leaving the little child to die from suffocation. It is indeed very sad which put responsibility on schools to find a solution to this problem. Parents are often worried about their children when they leave for school. Hence, they need detailed updates of their children movement and location.

7. Conclusions and Future Recommendations

7.1 Conclusions

Unfortunately, the work has not been hundred percent completed due to the COVID 19 challenge, yet it still accomplished over seventy percent of the system. With that progress, it was mainly divided between four main sub-systems; the bus, class, database, parents and user interface. When it came to the bus, the sub-system was able to recognize students with UHF RFID reader, create bus driver user interface, alarming when student did not leave the bus, parent user interface, speed limit for the driver, GSM messages for parents and school notifications, and also came up with the code of the operation. For the class, the sub-system was capable of recognizing students and save the information in an SD card, and GSM messages for students attending status. In addition, database on MySQL was created having a page, but with no further testing with Arduino. By having all of these progresses the only task left was to make the system communicate together. Since every user interface has been used with a Wi-Fi module either ESP32 or NodeMCU it had to exchange information with Arduino. Exchanging information can be done by serial communication or other communication protocols. In top of that database needed to be connected through Wi-Fi. In summary, most of

the work has been done except for communication between Arduino and Wi-Fi boards and further testing with room for improvements.

The project is unique by having a one system combining attendance, as well as bus monitoring. In addition, the system was capable of having a feature of recognizing students being left on the bus, which differentiate it with other similar projects. The feature was added with features that has been inspired from previous work. By creating such a project, safety awareness for bus problems can be spread by directing efforts towards the safety concern, which makes people learn and know about the safety issue. Sadly, the project was not able to be completed as the project component were not communicating. In other words, the ESP32, NodeMCU, and database were not exchanging information with the Arduino in one system.

During the process of the project, there were many different experiences, challenges, and lessons that improved us as engineers. One of the objectives was to learn about RFID application in solving real world problems. During the process of designing, we have experienced passive, UHF, and active RFID readers. In addition, there are many challenges that the system faced. With every step, the challenges expanded our knowledge and increased the problem-solving mentality with patience. One problem solving with patience lesson has been reflected was while working with the UHF RFID reader that needed to be integrated with Arduino. The reader had too many obstacles, where each obstacle taught a lesson. Moreover, the UHF RFID reader and active reader has opened our eyes into companies' SDKs and how to deal with the information given to create the desired output. Also, by having different readers with different SDK, an ability of evaluating the quality of documents have been obtained. Working in a team for almost a year has been very beneficial increasing the teamwork skills. In addition, critical thinking was involved while working. All in all, the unique project with inspiration of previous projects has improved the engineering skills of the group members teaching valuable lessons while trying to solve a safety issue.

7.2 Future Recommendations

Since there is always room for improvement and further development, this project can be improved by doing the following:

- **Use active RFID:** In our project we used passive RFID for the class and passive UHF RFID for the bus. The active RFID will give the advantage of reading for a wider range with more accurate reading results. More accurate readings with more distant will ensure the student safety even more making the project more powerful.
- **Uniformed user interface for all sub-systems:** In our project, each sub-system uses different applications to design the user interface which does not show consistency. The project will be more uniformed if the same application was used for all sub-systems.
- **Bigger database:** In our project, the database only includes attendance and absence information, and bus tacking information. However, it could be extended to have more information, such as student grades.
- **Add more classes:** Our project was only a prototype for one class and it could be improved to have more classes in the system.
- **Live stream on the bus:** This is a very good feature to add on the project to allow the parents to watch their children go to school and assure their safety. Also, it will record if any problem occurred in the bus.
- **Manual ID registering:** One of the project's constrains is the loss of students ID. This feature will allow registering the student ID manually making the system more flexible.

7.3 Future Work & Expected results

Since the project has not been fully completed, there were few tasks left. First, when it comes for attendance system, the RFID reader could be improved from passive to UHF passive RFID reader, or active RFID reader. The project was planned to upgrade the passive reader to UHF RFID to take attendance. The reader has been already acquired but needed some soldering as well as testing. By doing that attendance system can have a range advantage where it could automatically take attendance without students having to swipe their cards. Second, database required further testing as well as Wi-fi connection to Arduino. Third, Wi-Fi boards (ESP32, and NodeMCU) needed to have serial communication with the Arduino in order to exchange information. The communication could be also done by other communication protocols. Fourth, connecting, and testing the whole system together were not accomplished. Fifth, further cosmetic to tags and project was planned to be done as well. The tags could have printed pictures with names for each student. Also, the project could have been worked on to make it look as nice as possible. In addition, further video as well as banner describing the project would made the delivery of the idea better.

Completing the remaining tasks creates an expected reliable system that is capable of taking attendance as well as bus monitoring. The bus monitoring is expected to operate by having two modes that is controlled by the bus driver user interface to the ESP32 board. One will be arriving mode, and the other is starting mode as shown in Figures 6-7. The bus driver picks the right mode that will be then sent to the Arduino. When the Arduino receives the information, it will start counting and return back the students' name. It will also keep comparing the arrived students to the in-bus students to check if all students arrive. Also, it will display the students' absence, on bus, and arrived names as shown in Figures 8-9. The Arduino will have GSM send SMS notifications when necessary. Also, the Arduino would have the NodeMCU board that is connected to GPS to measure the speed of the bus and notify the Arduino if needed as well as updating the location. NodeMCU offers the parent user interface as shown in Figure 17. From Wi-Fi connection the Arduino will be capable of updating the database as well. The attendance system is expected to be able to detect students attending, being late, or absent by comparing time and further logic. Then, it will update the database with these information as well as notifying students' parents of the status of their children attendance via SMS. Both the bus monitoring as well as the class attendance updates one database making it easier for school to keep track of students. Both of the RFID readers would be UHF RFID reader taking the range advantage of over five meters to automate the process. The project architecture explains the flow of tasks shown in Figure 1.

8. References

Abd Wahab, Mohd Helmy & Abdul Kadir, Herdawatie & Yusof, Muhammad & Sanudin, Rahmat & Tomari, Razali. (2009). Class Attendance System using Active RFID: A Review.

Al-Lawati, A., Al-Jahdhami, S., Al-Belushi, A., Al-Adawi, D., Awadalla, M., & Al-Abri, D. (2015). RFID-based system for school children transportation safety enhancement. 2015 IEEE 8th GCC Conference & Exhibition. doi: 10.1109/ieeegcc.2015.7060047

Arab News. (2015). Probe launched into 'forgotten' student's death in Jeddah school bus. Retrieved from https://www.arabnews.com/saudi-arabia/news/820176

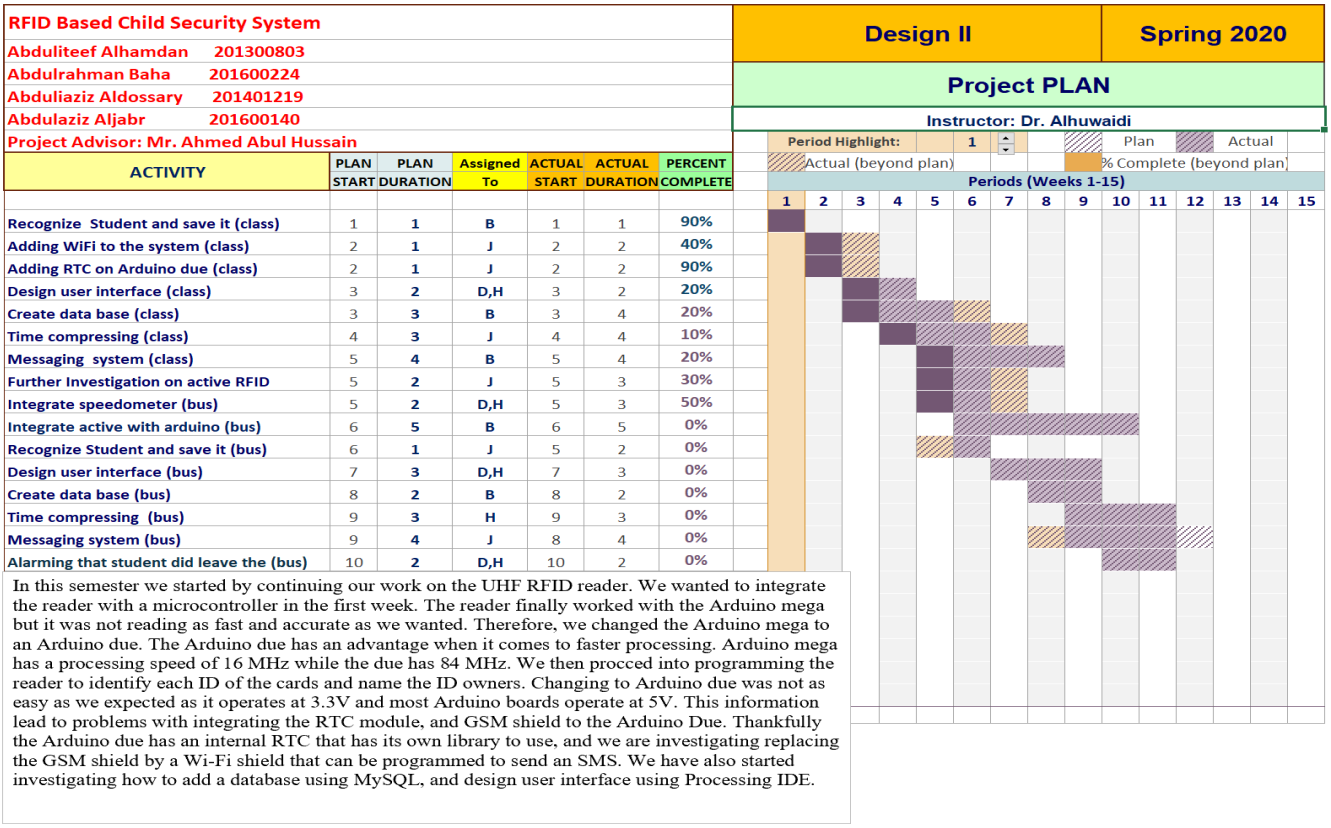
Asundkar, V. (2016). ENHANCE SAFETY SECURITY AND TRACKING SYSTEM FOR SCHOOL BUS AND CHILDREN. (2016). Retrieved from https://www.ijiert.org/admin/papers/1467308911_ICITER- 16 PUNE.PDF

Shaaban, K., Bekkali, A., Hamida, E. B., & Kadri, A. (2013). Smart Tracking System for School Buses Using Passive RFID Technology to Enhance Child Safety. *Journal of Traffic and Logistics Engineering*, *1*(2), 191–196. doi: 10.12720/jtle.1.2.191-196

Trad, S. (2018). Saudi student suffocates to death after being forgotten on school bus. Retrieved from https://stepfeed.com/saudi-student-suffocates-to-death-after-being-forgotten-on-school-bus-5838

Appendix A: Progress Reports

Progress report #1 in Figure 45



RFID Based Child Security System

Abduliteef Alhamdan 201300803
Abdulrahman Baha 201600224
Abduliaziz Aldossary 201401219
Abdulaziz Aljabr 201600140
Project Advisor: Mr. Ahmed Abul Hussain

Design II | Spring 2020

Project PLAN

Instructor: Dr. Alhuwaidi

ACTIVITY	PLAN START	PLAN DURATION	Assigned To	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Recognize Student and save it (class)	1	1	B	1	1	90%
Adding WiFi to the system (class)	2	1	J	2	2	40%
Adding RTC on Arduino due (class)	2	1	J	2	2	90%
Design user interface (class)	3	2	D,H	3	2	20%
Create data base (class)	3	3	B	3	4	20%
Time compressing (class)	4	3	J	4	4	10%
Messaging system (class)	5	4	B	5	4	20%
Further Investigation on active RFID	5	2	J	5	3	30%
Integrate speedometer (bus)	5	2	D,H	5	3	50%
Integrate active with arduino (bus)	6	5	B	6	5	0%
Recognize Student and save it (bus)	6	1	J	5	2	0%
Design user interface (bus)	7	3	D,H	7	3	0%
Create data base (bus)	8	2	B	8	2	0%
Time compressing (bus)	9	3	H	9	3	0%
Messaging system (bus)	9	4	J	8	4	0%
Alarming that student did leave the (bus)	10	2	D,H	10	2	0%

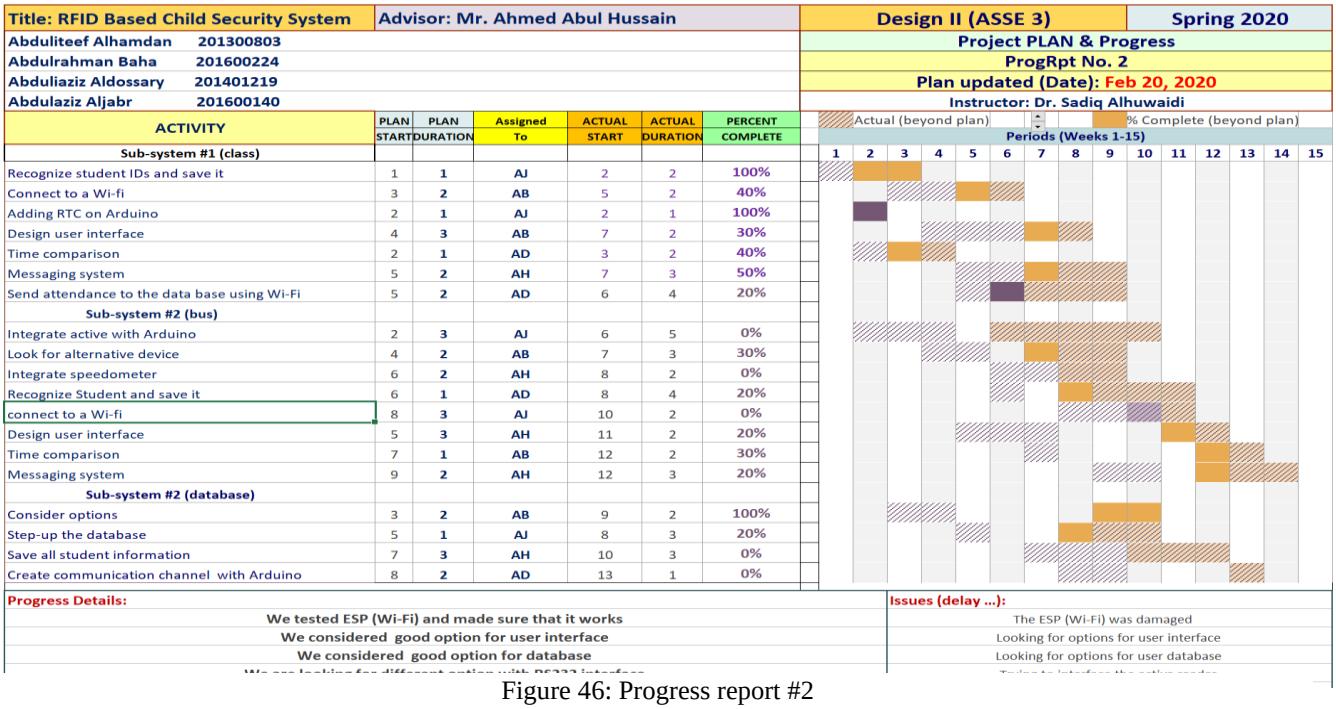
Period Highlight: 1 | Plan | Actual | Actual (beyond plan) | % Complete (beyond plan)

Periods (Weeks 1-15)

In this semester we started by continuing our work on the UHF RFID reader. We wanted to integrate the reader with a microcontroller in the first week. The reader finally worked with the Arduino mega but it was not reading as fast and accurate as we wanted. Therefore, we changed the Arduino mega to an Arduino due. The Arduino due has an advantage when it comes to faster processing. Arduino mega has a processing speed of 16 MHz while the due has 84 MHz. We then procced into programming the reader to identify each ID of the cards and name the ID owners. Changing to Arduino due was not as easy as we expected as it operates at 3.3V and most Arduino boards operate at 5V. This information lead to problems with integrating the RTC module, and GSM shield to the Arduino Due. Thankfully the Arduino due has an internal RTC that has its own library to use, and we are investigating replacing the GSM shield by a Wi-Fi shield that can be programmed to send an SMS. We have also started investigating how to add a database using MySQL, and design user interface using Processing IDE.

Figure 45: Progress report #1

Progress report #2 in Figure 46



Title: RFID Based Child Security System | Advisor: Mr. Ahmed Abul Hussain

Abduliteef Alhamdan 201300803
Abdulrahman Baha 201600224
Abduliaziz Aldossary 201401219
Abdulaziz Aljabr 201600140

Design II (ASSE 3) | Spring 2020

Project PLAN & Progress

ProgRpt No. 2

Plan updated (Date): Feb 20, 2020

Instructor: Dr. Sadiq Alhuwaidi

ACTIVITY	PLAN START	PLAN DURATION	Assigned To	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Sub-system #1 (class)						
Recognize student IDs and save it	1	1	AJ	2	2	100%
Connect to a Wi-fi	3	2	AB	5	2	40%
Adding RTC on Arduino	2	1	AJ	2	1	100%
Design user interface	4	3	AB	7	2	30%
Time comparison	2	1	AD	3	2	40%
Messaging system	5	2	AH	7	3	50%
Send attendance to the data base using Wi-Fi	5	2	AD	6	4	20%
Sub-system #2 (bus)						
Integrate active with Arduino	2	3	AJ	6	5	0%
Look for alternative device	4	2	AB	7	3	30%
Integrate speedometer	6	2	AH	8	2	0%
Recognize Student and save it	6	1	AD	8	4	20%
connect to a Wi-fi	8	3	AJ	10	2	0%
Design user interface	5	3	AH	11	2	20%
Time comparison	7	1	AB	12	2	30%
Messaging system	9	2	AH	12	3	20%
Sub-system #2 (database)						
Consider options	3	2	AB	9	2	100%
Step-up the database	5	1	AJ	8	3	20%
Save all student information	7	3	AH	10	3	0%
Create communication channel with Arduino	8	2	AD	13	1	0%

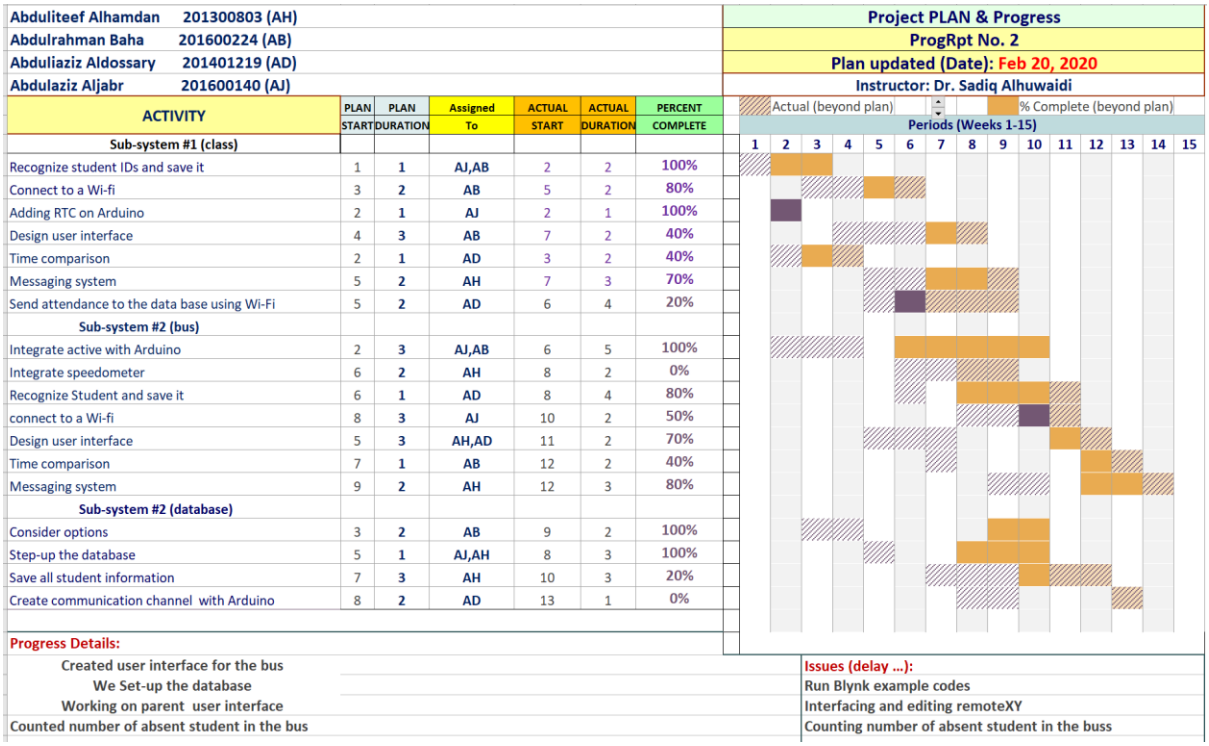
Actual (beyond plan) | % Complete (beyond plan)

Periods (Weeks 1-15)

Progress Details:	Issues (delay ...):
We tested ESP (Wi-Fi) and made sure that it works	The ESP (Wi-Fi) was damaged
We considered good option for user interface	Looking for options for user interface
We considered good option for database	Looking for options for user database

Figure 46: Progress report #2

Progress report #3 in Figure 47



Abduliteef Alhamdan	201300803 (AH)
Abdulrahman Baha	201600224 (AB)
Abduliaziz Aldossary	201401219 (AD)
Abdulaziz Aljabr	201600140 (AJ)

Project PLAN & Progress
ProgRpt No. 2
Plan updated (Date): Feb 20, 2020
Instructor: Dr. Sadiq Alhuwaidi

Actual (beyond plan) % Complete (beyond plan)
Periods (Weeks 1-15)

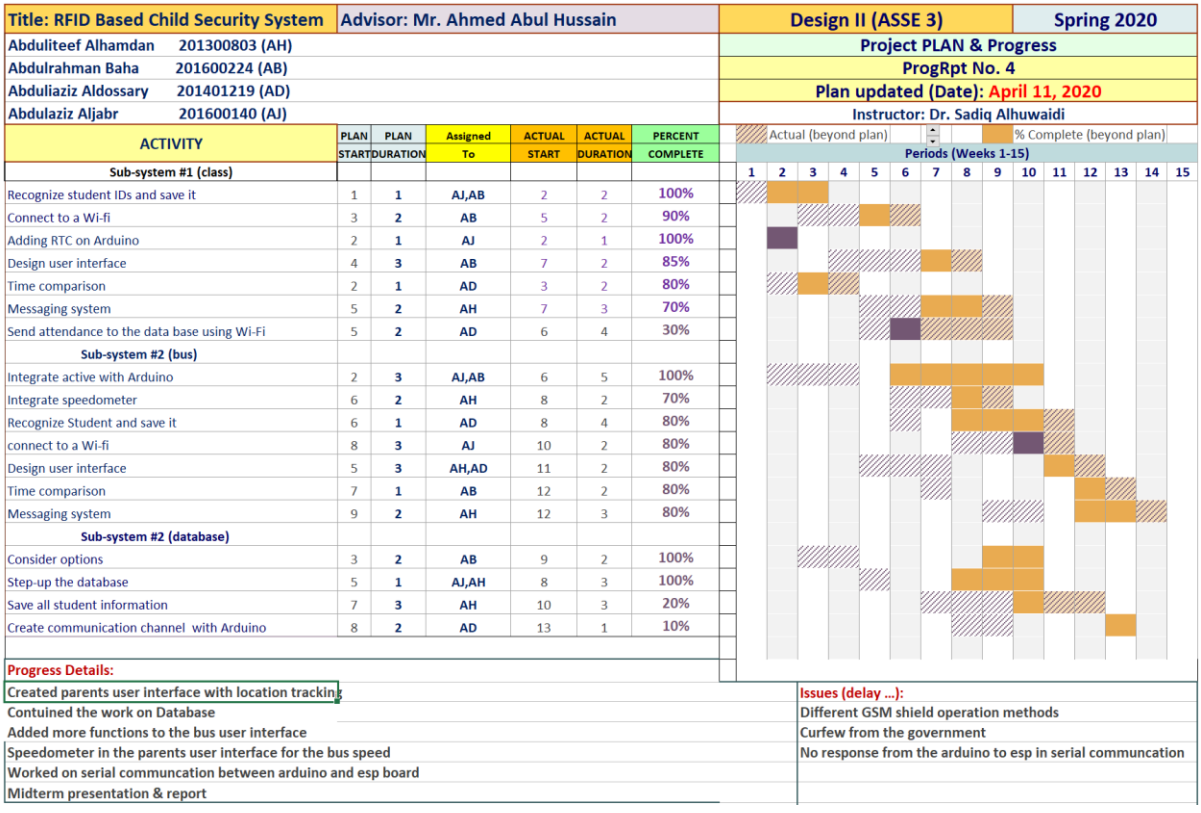
ACTIVITY	PLAN START	PLAN DURATION	Assigned To	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Sub-system #1 (class)						
Recognize student IDs and save it	1	1	AJ,AB	2	2	100%
Connect to a Wi-fi	3	2	AB	5	2	80%
Adding RTC on Arduino	2	1	AJ	2	1	100%
Design user interface	4	3	AB	7	2	40%
Time comparison	2	1	AD	3	2	40%
Messaging system	5	2	AH	7	3	70%
Send attendance to the data base using Wi-Fi	5	2	AD	6	4	20%
Sub-system #2 (bus)						
Integrate active with Arduino	2	3	AJ,AB	6	5	100%
Integrate speedometer	6	2	AH	8	2	0%
Recognize Student and save it	6	1	AD	8	4	80%
connect to a Wi-fi	8	3	AJ	10	2	50%
Design user interface	5	3	AH,AD	11	2	70%
Time comparison	7	1	AB	12	2	40%
Messaging system	9	2	AH	12	3	80%
Sub-system #2 (database)						
Consider options	3	2	AB	9	2	100%
Step-up the database	5	1	AJ,AH	8	3	100%
Save all student information	7	3	AH	10	3	20%
Create communication channel with Arduino	8	2	AD	13	1	0%

Progress Details:
Created user interface for the bus
We Set-up the database
Working on parent user interface
Counted number of absent student in the bus

Issues (delay ...):
Run Blynk example codes
Interfacing and editing remoteXY
Counting number of absent student in the buss

Figure 47: Progress report #3

Progress report #4 in Figure 48



Title: RFID Based Child Security System
Advisor: Mr. Ahmed Abul Hussain
Design II (ASSE 3)
Spring 2020

Abduliteef Alhamdan	201300803 (AH)
Abdulrahman Baha	201600224 (AB)
Abduliaziz Aldossary	201401219 (AD)
Abdulaziz Aljabr	201600140 (AJ)

Project PLAN & Progress
ProgRpt No. 4
Plan updated (Date): April 11, 2020
Instructor: Dr. Sadiq Alhuwaidi

Actual (beyond plan) % Complete (beyond plan)
Periods (Weeks 1-15)

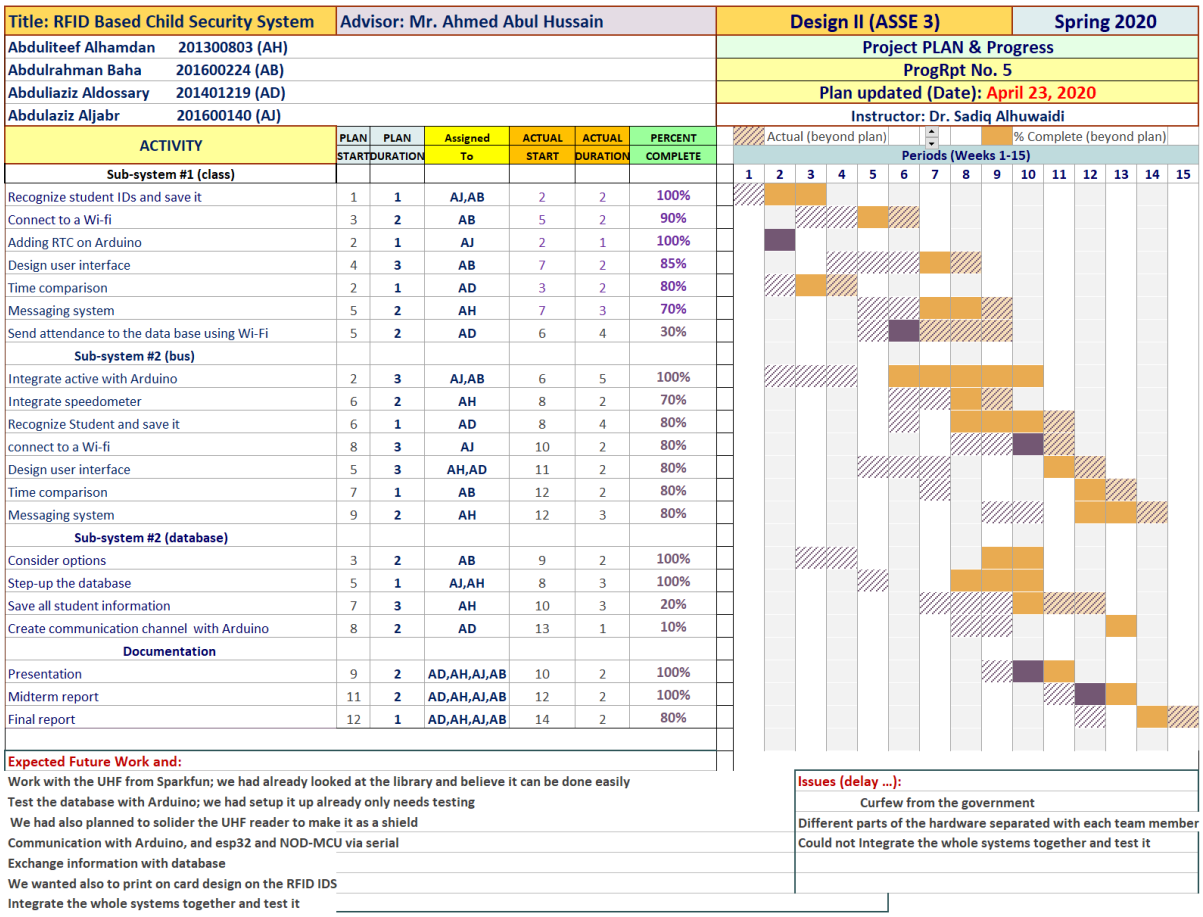
ACTIVITY	PLAN START	PLAN DURATION	Assigned To	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Sub-system #1 (class)						
Recognize student IDs and save it	1	1	AJ,AB	2	2	100%
Connect to a Wi-fi	3	2	AB	5	2	90%
Adding RTC on Arduino	2	1	AJ	2	1	100%
Design user interface	4	3	AB	7	2	85%
Time comparison	2	1	AD	3	2	80%
Messaging system	5	2	AH	7	3	70%
Send attendance to the data base using Wi-Fi	5	2	AD	6	4	30%
Sub-system #2 (bus)						
Integrate active with Arduino	2	3	AJ,AB	6	5	100%
Integrate speedometer	6	2	AH	8	2	70%
Recognize Student and save it	6	1	AD	8	4	80%
connect to a Wi-fi	8	3	AJ	10	2	80%
Design user interface	5	3	AH,AD	11	2	80%
Time comparison	7	1	AB	12	2	80%
Messaging system	9	2	AH	12	3	80%
Sub-system #2 (database)						
Consider options	3	2	AB	9	2	100%
Step-up the database	5	1	AJ,AH	8	3	100%
Save all student information	7	3	AH	10	3	20%
Create communication channel with Arduino	8	2	AD	13	1	10%

Progress Details:
Created parents user interface with location tracking
Contuined the work on Database
Added more functions to the bus user interface
Speedometer in the parents user interface for the bus speed
Worked on serial communcation between arduino and esp board
Midterm presentation & report

Issues (delay ...):
Different GSM shield operation methods
Curfew from the government
No response from the arduino to esp in serial communcation

Figure 48: Progress report #4

Progress report #5 in Figure 49



Title: RFID Based Child Security System	Advisor: Mr. Ahmed Abul Hussain	Design II (ASSE 3)	Spring 2020
Abduliteef Alhamdan 201300803 (AH)		Project PLAN & Progress	
Abdulrahman Baha 201600224 (AB)		ProgRpt No. 5	
Abduliaziz Aldossary 201401219 (AD)		Plan updated (Date): April 23, 2020	
Abdulaziz Aljabr 201600140 (AJ)		Instructor: Dr. Sadiq Alhuwaidi	

ACTIVITY	PLAN START	PLAN DURATION	Assigned To	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Sub-system #1 (class)						
Recognize student IDs and save it	1	1	AJ,AB	2	2	100%
Connect to a Wi-fi	3	2	AB	5	2	90%
Adding RTC on Arduino	2	1	AJ	2	1	100%
Design user interface	4	3	AB	7	2	85%
Time comparison	2	1	AD	3	2	80%
Messaging system	5	2	AH	7	3	70%
Send attendance to the data base using Wi-Fi	5	2	AD	6	4	30%
Sub-system #2 (bus)						
Integrate active with Arduino	2	3	AJ,AB	6	5	100%
Integrate speedometer	6	2	AH	8	2	70%
Recognize Student and save it	6	1	AD	8	4	80%
connect to a Wi-fi	8	3	AJ	10	2	80%
Design user interface	5	3	AH,AD	11	2	80%
Time comparison	7	1	AB	12	2	80%
Messaging system	9	2	AH	12	3	80%
Sub-system #2 (database)						
Consider options	3	2	AB	9	2	100%
Step-up the database	5	1	AJ,AH	8	3	100%
Save all student information	7	3	AH	10	3	20%
Create communication channel with Arduino	8	2	AD	13	1	10%
Documentation						
Presentation	9	2	AD,AH,AJ,AB	10	2	100%
Midterm report	11	2	AD,AH,AJ,AB	12	2	100%
Final report	12	1	AD,AH,AJ,AB	14	2	80%

Expected Future Work and:

Work with the UHF from Sparkfun; we had already looked at the library and believe it can be done easily
Test the database with Arduino; we had setup it up already only needs testing
We had also planned to solider the UHF reader to make it as a shield
Communication with Arduino, and esp32 and NOD-MCU via serial
Exchange information with database
We wanted also to print on card design on the RFID IDS
Integrate the whole systems together and test it

Issues (delay ...):

Curfew from the government
Different parts of the hardware separated with each team member
Could not Integrate the whole systems together and test it

Figure 49: Progress report #5

Appendix B: Program Codes

1) Arduino code for Bus user interface programmed via Arduino IDE to ESP32:

```
#define REMOTEXY_MODE__ESP32CORE_WIFI_POINT
#include <WiFi.h>
#include <RemoteXY.h>
// RemoteXY connection settings
#define REMOTEXY_WIFI_SSID "RemoteXY"
#define REMOTEXY_WIFI_PASSWORD ""
#define REMOTEXY_SERVER_PORT 6377
// RemoteXY configurate
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
  { 255,2,0,114,1,36,2,10,13,4,
  65,6,19,35,9,9,3,65,6,19,
  53,9,9,3,129,0,0,21,44,6,
  3,17,84,114,105,112,32,105,110,100,
  105,99,97,116,111,114,115,58,0,131,
  0,46,2,36,6,1,2,31,80,114,
  101,115,101,110,116,32,83,116,117,100,
  101,110,116,115,0,131,0,46,14,36,
  6,2,2,31,65,98,115,101,110,116,
```

```
  32,83,116,117,100,101,110,116,115,0,
  131,1,43,44,56,15,3,2,31,77,
  97,105,110,32,80,97,103,101,0,67,
  4,3,3,39,5,1,2,26,11,67,
  4,3,9,39,5,1,2,26,11,67,
  4,3,15,39,5,1,2,26,11,67,
  4,3,21,39,5,1,2,26,11,67,
  4,3,27,39,5,1,2,26,11,67,
  4,3,33,39,5,1,2,26,11,67,
  4,3,39,39,5,1,2,26,11,67,
  4,3,45,39,5,1,2,26,11,67,
  4,3,51,39,5,1,2,26,11,67,
  4,3,57,39,5,1,2,26,11,67,
  4,3,3,39,5,2,2,26,11,67,
  4,3,9,39,5,2,2,26,11,67,
  4,3,15,39,5,2,2,26,11,67,
  4,3,21,39,5,2,2,26,11,67,
  4,3,27,39,5,2,2,26,11,67,
  4,3,33,39,5,2,2,26,11,67,
  4,3,39,39,5,2,2,26,11,67,
  4,3,45,39,5,2,2,26,11,67,
  4,3,51,39,5,2,2,26,11,67,
  4,3,57,39,5,2,2,26,11,2,
  1,1,36,14,6,3,2,26,31,31,
  79,78,0,79,70,70,0,2,1,1,
  53,14,6,3,2,26,31,31,79,78,
  0,79,70,70,0,129,0,1,29,18,
  6,3,17,83,116,97,114,116,0,129,
  0,1,46,18,6,3,17,65,114,114,
  105,118,97,108,0,69,1,36,31,10,
  10,3,66,0,85,1,7,8,0,2,
  26,66,0,85,12,7,9,0,2,26,
  67,5,3,2,36,11,3,2,26,31,
  131,0,46,25,36,6,4,2,31,65,
  114,114,105,118,101,100,32,83,116,117,
  100,101,110,116,115,0,66,0,85,23,
  7,9,0,2,26,67,4,3,3,39,
  5,4,2,26,11,67,4,3,9,39,
  5,4,2,26,11,67,4,3,15,39,
  5,4,2,26,11,67,4,3,21,39,
  5,4,2,26,11,67,4,3,27,39,
  5,4,2,26,11,67,4,3,33,39,
  5,4,2,26,11,67,4,3,39,39,
  5,4,2,26,11,67,4,3,45,39,
  5,4,2,26,11,67,4,3,51,39,
  5,4,2,26,11,67,4,3,57,39,
  5,4,2,26,11 };
// this structure defines all the variables and events of your control interface
struct {
    // input variables
  uint8_t switch_S; // =1 if switch ON and =0 if OFF
  uint8_t switch_A; // =1 if switch ON and =0 if OFF
    // output variables
  uint8_t led_1_r; // =0..255 LED Red brightness
  uint8_t led_1_g; // =0..255 LED Green brightness
  uint8_t led_2_r; // =0..255 LED Red brightness
  uint8_t led_2_g; // =0..255 LED Green brightness
  char text_1P[11];  // string UTF8 end zero
  char text_2P[11];  // string UTF8 end zero
  char text_3P[11];  // string UTF8 end zero
```

```
  char text_4P[11];  // string UTF8 end zero
  char text_5P[11];  // string UTF8 end zero
  char text_6P[11];  // string UTF8 end zero
  char text_7P[11];  // string UTF8 end zero
  char text_8P[11];  // string UTF8 end zero
  char text_9P[11];  // string UTF8 end zero
  char text_10P[11];  // string UTF8 end zero
  char text_1A[11];  // string UTF8 end zero
  char text_2A[11];  // string UTF8 end zero
  char text_3A[11];  // string UTF8 end zero
  char text_4A[11];  // string UTF8 end zero
  char text_5A[11];  // string UTF8 end zero
  char text_6A[11];  // string UTF8 end zero
  char text_7A[11];  // string UTF8 end zero
  char text_8A[11];  // string UTF8 end zero
  char text_9A[11];  // string UTF8 end zero
  char text_10A[11];  // string UTF8 end zero
  int16_t sound_1; // =0 no sound, else ID of sound, =1001 for example, look sound list in app
  int8_t level_1; // =0..100 level position
  int8_t level_2; // =0..100 level position
  char text_12[11];  // string UTF8 end zero
  int8_t level_3; // =0..100 level position
  char text_1Ar[11];  // string UTF8 end zero
  char text_2Ar[11];  // string UTF8 end zero
  char text_3Ar[11];  // string UTF8 end zero
  char text_4Ar[11];  // string UTF8 end zero
  char text_5Ar[11];  // string UTF8 end zero
  char text_6Ar[11];  // string UTF8 end zero
  char text_7Ar[11];  // string UTF8 end zero
  char text_8Ar[11];  // string UTF8 end zero
  char text_9Ar[11];  // string UTF8 end zero
  char text_10Ar[11];  // string UTF8 end zero
    // other variable
  uint8_t connect_flag;  // =1 if wire connected, else =0
} RemoteXY;
#pragma pack(pop)
int i;
int ar;
int at;
   String student;
int arival; int arv=0;
int start; int st=0;
int c1=0; int c2=0; int c3=0; int c4=0; int c5=0; int c6=0; int c7=0; int c8=0; int c9=0; int c10=0;
int   ab=10;
 String Students[11]={"Fahad", "Saad", "Ali", "Omar", "Saleh","Ahmed", "Salem", "Nawaf", "Khaled",
"Yousef", "           "};
 String y;
void setup()
{
  RemoteXY_Init ();

  // TODO you setup code
}
void loop()
{
  RemoteXY_Handler ();
  student="Saad";//testing
 ar=1;
ab=9;
at=2;
```

```
Nameat();
Namear();

Nameabs();
 // for color of modes
  if (RemoteXY.switch_S ==0) {
    /*  switch on  */
   RemoteXY.led_1_r=255;
   RemoteXY.led_1_g=0;

  }
  else if (RemoteXY.switch_S ==1 && st==0 ){
    /*  on  */
   RemoteXY.led_1_r=0;
   RemoteXY.led_1_g=255;
   start=1;
   arival=0;
   arv=0;
   st=1;
  c1=0; c2=0;c3=0;c4=0;c5=0;c6=0;c7=0;c8=0;c9=0;c10=0;
  }
   if (RemoteXY.switch_A ==0) {
    /*  switch on  */

    RemoteXY.led_2_r=255;
   RemoteXY.led_2_g=0;

  }
  else if ( RemoteXY.switch_A ==1&& arv==0)  {
    /*  switch off  */

   RemoteXY.led_2_r=0;
   RemoteXY.led_2_g=255;
    arv=1;
   st=0;
   start=0;
arival=1;
c1=0; c2=0;c3=0;c4=0;c5=0;c6=0;c7=0;c8=0;c9=0;c10=0;
  }
  //for mode indicatuion
  if ( (RemoteXY.switch_S ==0 && RemoteXY.switch_A ==0) || ( RemoteXY.switch_S ==1 &&
RemoteXY.switch_A ==1) ) {

     strcpy  (RemoteXY.text_12, "Error");

  }
  if  (RemoteXY.switch_S ==1 && RemoteXY.switch_A ==0)
  {
     strcpy  (RemoteXY.text_12, "Start");
  }
   if  (RemoteXY.switch_S ==0 && RemoteXY.switch_A ==1)
  {
     strcpy  (RemoteXY.text_12, "Arrival");
  }
  if (ar<at){
   RemoteXY.sound_1=2003;
  }
}

  void Nameat(){
```

```
  //for registering names attended

  switch(at){
    case 1:

    sprintf  (RemoteXY.text_1P, "%s", student);
    RemoteXY.level_1 = 10;
  break;
  case 2:
    sprintf (RemoteXY.text_2P, "%s", student);
    RemoteXY.level_1 = 20;
  break;
  case 3:
    sprintf  (RemoteXY.text_3P, "%s", student);
    RemoteXY.level_1 = 30;
  break;
  case 4:
   sprintf  (RemoteXY.text_4P, "%s", student);
    RemoteXY.level_1 = 40;
  break;
  case 5:
   sprintf  (RemoteXY.text_5P, "%s", student);
    RemoteXY.level_1 = 50;
  break;
  case 6:
    sprintf (RemoteXY.text_6P, "%s", student);
    RemoteXY.level_1 = 60;
  break;
  case 7:
     sprintf  (RemoteXY.text_7P, "%s", student);
    RemoteXY.level_1 = 70;
  break;
  case 8:
     sprintf (RemoteXY.text_8P, "%s", student);
    RemoteXY.level_1 = 80;
  break;
  case 9:
    sprintf (RemoteXY.text_9P, "%s", student);
    RemoteXY.level_1 = 90;
  break;
  case 10:
sprintf  (RemoteXY.text_10P, "%s", student);
    RemoteXY.level_1 = 100;
  break;
  default:
   RemoteXY.level_1 = 0;
  break;
  }
 }

   void Namear(){

  switch(ar){
    case 1:

    sprintf  (RemoteXY.text_1Ar, "%s", student);
    RemoteXY.level_1 = 10;
  break;
  case 2:
    sprintf (RemoteXY.text_2Ar, "%s", student);
```

```
    RemoteXY.level_1 = 20;
   break;
   case 3:
     sprintf  (RemoteXY.text_3Ar, "%s", student);
     RemoteXY.level_1 = 30;
   break;
   case 4:
    sprintf  (RemoteXY.text_4Ar, "%s", student);
     RemoteXY.level_1 = 40;
   break;
   case 5:
    sprintf  (RemoteXY.text_5Ar, "%s", student);
     RemoteXY.level_1 = 50;
   break;
   case 6:
     sprintf (RemoteXY.text_6Ar, "%s", student);
     RemoteXY.level_1 = 60;
   break;
   case 7:
      sprintf  (RemoteXY.text_7Ar, "%s", student);
     RemoteXY.level_1 = 70;
   break;
   case 8:
      sprintf (RemoteXY.text_8Ar, "%s", student);
     RemoteXY.level_1 = 80;
   break;
   case 9:
     sprintf (RemoteXY.text_9Ar, "%s", student);
     RemoteXY.level_1 = 90;
   break;
   case 10:
sprintf  (RemoteXY.text_10Ar, "%s", student);
     RemoteXY.level_1 = 100;
   break;

   default:
    RemoteXY.level_1 = 0;
   break;
   }
 }

 void Nameabs(){

  switch(ab){

 case 10:
     sprintf (RemoteXY.text_1A, "%s", Students[0]);
      sprintf (RemoteXY.text_2A, "%s", Students[1]);
       sprintf (RemoteXY.text_3A, "%s", Students[2]);
        sprintf (RemoteXY.text_4A, "%s", Students[3]);
         sprintf (RemoteXY.text_5A, "%s", Students[4]);
          sprintf (RemoteXY.text_6A, "%s", Students[5]);
           sprintf (RemoteXY.text_7A, "%s", Students[6]);
            sprintf (RemoteXY.text_8A, "%s", Students[7]);
             sprintf (RemoteXY.text_9A, "%s", Students[8]);
              sprintf (RemoteXY.text_10A, "%s", Students[9]);

    RemoteXY.level_2 = 100;
   break;
   case 9:
```

```
  Searchrep();
Students[9]=Students[10];
    sprintf (RemoteXY.text_1A, "%s", Students[0]);
     sprintf (RemoteXY.text_2A, "%s", Students[1]);
      sprintf (RemoteXY.text_3A, "%s", Students[2]);
       sprintf (RemoteXY.text_4A, "%s", Students[3]);
        sprintf (RemoteXY.text_5A, "%s", Students[4]);
         sprintf (RemoteXY.text_6A, "%s", Students[5]);
          sprintf (RemoteXY.text_7A, "%s", Students[6]);
           sprintf (RemoteXY.text_8A, "%s", Students[7]);
            sprintf (RemoteXY.text_9A, "%s", Students[8]);


    RemoteXY.level_2 = 90;
  break;
  case 8:
  Searchrep();
 Students[8]=Students[10];
    sprintf (RemoteXY.text_1A, "%s", Students[0]);
     sprintf (RemoteXY.text_2A, "%s", Students[1]);
      sprintf (RemoteXY.text_3A, "%s", Students[2]);
       sprintf (RemoteXY.text_4A, "%s", Students[3]);
        sprintf (RemoteXY.text_5A, "%s", Students[4]);
         sprintf (RemoteXY.text_6A, "%s", Students[5]);
          sprintf (RemoteXY.text_7A, "%s", Students[6]);
           sprintf (RemoteXY.text_8A, "%s", Students[7]);


    RemoteXY.level_2 = 80;
  break;
  case 7:
  Searchrep();
 Students[7]=Students[10];
   sprintf (RemoteXY.text_1A, "%s", Students[0]);
     sprintf (RemoteXY.text_2A, "%s", Students[1]);
      sprintf (RemoteXY.text_3A, "%s", Students[2]);
       sprintf (RemoteXY.text_4A, "%s", Students[3]);
        sprintf (RemoteXY.text_5A, "%s", Students[4]);
         sprintf (RemoteXY.text_6A, "%s", Students[5]);
          sprintf (RemoteXY.text_7A, "%s", Students[6]);


    RemoteXY.level_2 = 70;
  break;
  case 6:
  Searchrep();
 Students[6]=Students[10];
    sprintf (RemoteXY.text_1A, "%s", Students[0]);
     sprintf (RemoteXY.text_2A, "%s", Students[1]);
      sprintf (RemoteXY.text_3A, "%s", Students[2]);
       sprintf (RemoteXY.text_4A, "%s", Students[3]);
        sprintf (RemoteXY.text_5A, "%s", Students[4]);
         sprintf (RemoteXY.text_6A, "%s", Students[5]);

    RemoteXY.level_2 = 60;
  break;
  case 5:
  Searchrep();
 Students[5]=Students[10];
  sprintf (RemoteXY.text_1A, "%s", Students[0]);
```

```
    sprintf (RemoteXY.text_2A, "%s", Students[1]);
     sprintf (RemoteXY.text_3A, "%s", Students[2]);
      sprintf (RemoteXY.text_4A, "%s", Students[3]);
       sprintf (RemoteXY.text_5A, "%s", Students[4]);


   RemoteXY.level_2 = 50;
  break;
  case 4:
  Searchrep();
 Students[4]=Students[10];
     sprintf (RemoteXY.text_1A, "%s", Students[0]);
    sprintf (RemoteXY.text_2A, "%s", Students[1]);
     sprintf (RemoteXY.text_3A, "%s", Students[2]);
      sprintf (RemoteXY.text_4A, "%s", Students[3]);



   RemoteXY.level_2 = 40;
  break;
  case 3:

  Searchrep();
 Students[3]=Students[10];
sprintf (RemoteXY.text_1Ar, "%s", Students[0]);
    sprintf (RemoteXY.text_2A, "%s", Students[1]);
     sprintf (RemoteXY.text_3A, "%s", Students[2]);



   RemoteXY.level_2 = 30;
  break;
  case 2:
  Searchrep();
 Students[2]=Students[10];
     sprintf (RemoteXY.text_1A, "%s", Students[0]);
    sprintf (RemoteXY.text_2A, "%s", Students[1]);



   RemoteXY.level_2 = 20;
  break;
  case 1:
  Searchrep();
 Students[1]=Students[10];
       sprintf (RemoteXY.text_1Ar, "%s", Students[0]);

   case 0:
    sprintf (RemoteXY.text_1A, "%s", Students[10]);
    sprintf (RemoteXY.text_2A, "%s", Students[10]);
     sprintf (RemoteXY.text_3A, "%s", Students[10]);
      sprintf (RemoteXY.text_4A, "%s", Students[10]);
       sprintf (RemoteXY.text_5A, "%s", Students[10]);
        sprintf (RemoteXY.text_6A, "%s", Students[10]);
         sprintf (RemoteXY.text_7A, "%s", Students[10]);
          sprintf (RemoteXY.text_8A, "%s", Students[10]);
           sprintf (RemoteXY.text_9A, "%s", Students[10]);
            sprintf (RemoteXY.text_10A, "%s", Students[10]);
```

```
    RemoteXY.level_2 = 0;
   break;
   default:
    RemoteXY.level_2 = 0;
   break;
   }
  }
 void Searchrep(){
   //aray fix needed
  // to search then replace the names for absent Studentss
   for(i=1; i<=10; i++){
if(student == Students[i]){
y=Students[i];
Students[i]=Students[ab];
Students[ab]=y;
break;
}

}
  }
```

2) UHF RFID student recognition & bus counting logic code to Arduino Due:

```
#include "Arduino.h"
#include <avr/pgmspace.h>
#define SW 8 //testing replace with userinterface in future
byte ab =10;  byte ar=0;  byte arvi=0; byte start=0; byte at=0; byte arv=0; byte St=0;
const char Fahad[]  PROGMEM = {0x27, 0x40, 0x07, 0x87};
const char Saad[]   PROGMEM = {0x11, 0x0, 0xA9, 0xCE};
const char Ali[]    PROGMEM = {0x18, 0x80, 0x56, 0xA1};
const char Omar[]   PROGMEM = {0x27, 0x70, 0x07, 0x09};
const char Saleh[]  PROGMEM = {0x27, 0x60, 0x08, 0x33};
const char Ahmed[]  PROGMEM = {0x27,0x50, 0x06, 0xB8};
const char Salem[]  PROGMEM = {0x88,0x99, 0xAA, 0xBC};
const char Nawaf[]  PROGMEM = {0x27, 0x10, 0x8, 0xCD};
const char Khaled[] PROGMEM = {0x27, 0x10, 0x09, 0x56};
const char Yousef[] PROGMEM = {0x28, 0x30, 0x02, 0xF0};
const char RdTagAM2[] PROGMEM
={0x53,0x57,0x00,0x0A,0xFF,0x02,0x01,0x02,0x06,0x00,0x00,0x00,0x00,0x42};
const char IDcmd[]    PROGMEM ={0x53,0x57,0x00,0x03,0xFF,0x20,0x34};

void ReadTag_AM2()
{
  Serial1.write(RdTagAM2,14);
}

unsigned char incomingByte;
byte response[100];
byte ID[20];
void sendIdentifyCmd ()
{
  Serial1.write(IDcmd,7);
}

void setup ()
{
   pinMode(SW,INPUT);
  Serial1.begin (9600);
  Serial.begin (9600);

  delay(1000);
```

```
  //Serial1.flush();
   Serial.println ("begin initial Serial1!\n");
  delay(1000);
   sendIdentifyCmd ();
  delay (2);
    while(Serial1.available () > 0)
  {

     incomingByte=Serial1.read ();;
     Serial.print (incomingByte,HEX);
     Serial.print (' ');
  }

  }
boolean fc1=false; boolean fc2=false; boolean fc3=false; boolean fc4=false; boolean fc5=false;
boolean fc6=false; boolean fc7=false; boolean fc8=false; boolean fc9=false; boolean fc10=false;

void loop ()
{
  int index=0;
  Serial1.flush();
  ReadTag_AM2(); delay(100);

  while(Serial1.available () > 0) // read and print the full response
  {
     incomingByte=Serial1.read ();
     response[index]=incomingByte;
     Serial.print (response[index],HEX);
    // Serial.print ('*');
     index++;
  }
  Serial1.flush();
  Serial.println ("read id");
  delay(40);
  Serial.print("i="); Serial.println(index);
  if (index==20)
  { byte k=0; //reset index
  Serial.println("checking id");
   for (byte j=15;j<19;j++)  //extract and print only the tag ID
    {
    ID[k]=response[j];
   Serial.print (ID[k],HEX);
    //Serial.println ("all 20d");
  //  Serial.print(pgm_read_byte_near(Omar + k),HEX);
    k++;
    }
  }
  else // reset values to 0
  {
   byte k=0; //reset index
   for (byte j=26;j<30;j++)  //extract and print only the tag ID
    {
    ID[k]=0;
    k++;
    }
  }
boolean SWVal = 0; //digitalRead(SW); // at pin8
if (SWVal==0 && St==0)
{
  start=1;
```

```
  arvi=0;
  St=1;
 arv=0;
  fc1=false; fc2=false;fc3=false;fc4=false;fc5=false;
  fc6=false;fc7=false;fc8=false;fc9=false;fc10=false;
  Serial.print("Start Mode");
}

else if (SWVal==1 && arv==0){
  start=0;
  arvi=1;
  arv=1;
  St=0;
  fc1=false; fc2=false;fc3=false;fc4=false;fc5=false;
  fc6=false;fc7=false;fc8=false;fc9=false;fc10=false;
  Serial.print("Arrival Mode");
}
if( (ID[0] == pgm_read_byte_near(Fahad + 0)) && (ID[1] == pgm_read_byte_near(Fahad + 1)) && (ID[2] ==
pgm_read_byte_near(Fahad + 2)) && (ID[3] == pgm_read_byte_near(Fahad + 3)))
        {
        Serial.println("id matched Fahd");

         if (!fc1)
         {
         if(arvi==1){
           ar+=1;
           Serial.println("Arrived student Fahad +1");
         //  Namear();
         }
        else if(start==1){
          at+=1;
            ab-=1;
             Serial.println("Attended student & absent Fahad +1 -1");
        //  Nameat();
         }

         fc1=true;
        // verifyCheckIn();
         Serial.println("Fahad here");
        }     // else fc1=false;
        }
        else if( (ID[0] == pgm_read_byte_near(Saad + 0)) && (ID[1] == pgm_read_byte_near(Saad + 1)) &&
(ID[2] == pgm_read_byte_near(Saad + 2)) && (ID[3] == pgm_read_byte_near(Saad + 3)))
        {
         Serial.println("id matched Saad");
         if (fc2==false)
         {
             if (arvi==1){
           ar+=1;
             Serial.println("Arrived student Saad +1");
         //  Namear();
         }
         else if(start==1){
          at+=1;
            ab-=1;
             Serial.println("Attended student & absent Saad +1 -1");
        //  Nameat();
         }
         fc2=true;
         //verifyCheckIn();
```

```
        Serial.println("Saad here");
       }
        //else fc2=false;

       }
        else if( (ID[0] == pgm_read_byte_near(Ali + 0)) && (ID[1] == pgm_read_byte_near(Ali + 1)) && (ID[2]
== pgm_read_byte_near(Ali + 2)) && (ID[3] == pgm_read_byte_near(Ali + 3)))

       {          Serial.println("id matched Ali");
        if (fc3==false)
        {
        if (arvi==1){
          ar+=1;
            Serial.println("Arivved student Ali +1");
        //  Namear();
        }
        else if(start==1){
         at+=1;
           ab-=1;
            Serial.println("Attended student & absent Ali +1 -1");
       //  Nameat();
        }

        fc3=true;
       // verifyCheckIn();
        Serial.println("Ali here");
       }    //    else fc3=false;
       }
       else if( (ID[0] == pgm_read_byte_near(Omar + 0)) && (ID[1] == pgm_read_byte_near(Omar + 1)) &&
(ID[2] == pgm_read_byte_near(Omar + 2)) && (ID[3] == pgm_read_byte_near(Omar + 3)))

       {          Serial.println("id matched Omar");
          if (fc4==false)
          {
            if (arvi==1){
          ar+=1;
            Serial.println("Arivved student Omar +1");
        //  Namear();
        }
        else if(start==1){
          at+=1;
            ab-=1;
             Serial.println("Attended student & absent Omar +1 -1");
       //  Nameat();
        }

        fc4=true;
       // verifyCheckIn();
        Serial.println("Omar here");
       }    //    else fc4=false;

       }
       else if( (ID[0] == pgm_read_byte_near(Saleh + 0)) && (ID[1] == pgm_read_byte_near(Saleh + 1)) &&
(ID[2] == pgm_read_byte_near(Saleh + 2)) && (ID[3] == pgm_read_byte_near(Saleh + 3)))

       {
        Serial.println("saleh id matched");
        if (fc5==false)
       {
            if (arvi==1){
```

```
          ar+=1;
            Serial.println("Arivved student Saleh +1");
         //  Namear();
         }
         else if(start==1){
          at+=1;
            ab-=1;
             Serial.println("Attended student & absent Saleh +1 -1");
        //  Nameat();
         }

         fc5=true;
         Serial.print("Saleh here");
        }     //   else fc5=false;

        }
       else if( (ID[0] == pgm_read_byte_near(Ahmed + 0)) && (ID[1] == pgm_read_byte_near(Ahmed + 1)) &&
(ID[2] == pgm_read_byte_near(Ahmed + 2)) && (ID[3] == pgm_read_byte_near(Ahmed + 3)))

        {          Serial.println("id matched Ahmed");
         if (fc6==false)
         {
            if (arvi==1){
          ar+=1;
            Serial.println("Arivved student Ahmed +1");
         //  Namear();
         }
         else if(start==1){
          at+=1;
        //  Nameat();
          ab-=1;
           Serial.println("Attended student & absent Ahmed +1 -1");
         }

         fc6=true;
         //verifyCheckIn();
         Serial.println("Ahmed here");
        }  //      else fc6=false;

        }

      else if( (ID[0] == pgm_read_byte_near(Salem + 0)) && (ID[1] == pgm_read_byte_near(Salem + 1)) &&
(ID[2] == pgm_read_byte_near(Salem + 2)) && (ID[3] == pgm_read_byte_near(Salem + 3)))
        {          Serial.println("id matched Salem");
           if (fc7==false)
           {
             if (arvi==1){
           ar+=1;
            Serial.println("Arivved student Salem +1");
         //  Namear();
         }
        else if(start==1){
          at+=1;
        //  Nameat();
          ab-=1;
           Serial.println("Attended student & absent Salem +1 -1");
         }

         fc7=true;
        // verifyCheckIn();
```

```
 Serial.println("Salem here");

}     //    else fc7=false;

}
else if( (ID[0] == pgm_read_byte_near(Nawaf + 0)) && (ID[1] == pgm_read_byte_near(Nawaf + 1)) &&
(ID[2] == pgm_read_byte_near(Nawaf + 2)) && (ID[3] == pgm_read_byte_near(Nawaf + 3)))

{
          Serial.println("id matched Nawaf");
   if (fc8==false)
   {
     if(arvi==1){
   ar+=1;
 //  Namear();
   Serial.println("Arivved student Nawaf +1");
 }
 else if(start==1){
  at+=1;
    ab-=1;
     Serial.println("Attended student & absent Nawaf +1 -1");
 //  Nameat();
 }

 fc8=true;
 // verifyCheckIn();
 Serial.println("Nawaf here");
}    //     else fc8=false;
}
else if( (ID[0] == pgm_read_byte_near(Khaled + 0)) && (ID[1] == pgm_read_byte_near(Khaled + 1)) &&
(ID[2] == pgm_read_byte_near(Khaled + 2)) && (ID[3] == pgm_read_byte_near(Khaled + 3)))

{          Serial.println("id matched Khaled");
   if (fc9==false)
   {
     if (arvi==1){
   ar+=1;
     Serial.println("Arivved student Khaled +1");
 //  Namear();
 }
 else if(start==1){
  at+=1;
    ab-=1;
     Serial.println("Attended student & absent Khaled +1 -1");
 //  Nameat();
 }
      fc9=true;
 // verifyCheckIn();
 Serial.println("Khaled here");
}   //      else fc9=false;
}
else if( (ID[0] == pgm_read_byte_near(Yousef + 0)) && (ID[1] == pgm_read_byte_near(Yousef + 1)) &&
(ID[2] == pgm_read_byte_near(Yousef + 2)) && (ID[3] == pgm_read_byte_near(Yousef + 3)))

{          Serial.println("id matched Yusuf");
 if (fc10==false)
 {
 if (arvi==1){
  ar+=1;
    Serial.println("Arivved student Yousef+1");
```

```
        //  Namear();
         }
        else if(start==1){
          at+=1;
            ab-=1;
             Serial.println("Attended student & absent Yosef +1 -1");
        //  Nameat();
         }

         fc10=true;
        // verifyCheckIn();
         Serial.println("Yousef here");
       }
       }
       else {
         Serial.println("not registered");
       }
       Serial.print("Arrival: "); Serial.print(arvi); Serial.print("Start: "); Serial.println(start);
       Serial.print(fc1);Serial.print(" "); Serial.print(fc2);Serial.print(" ");Serial.print(fc3);Serial.print("
");Serial.print(fc4);Serial.print(" ");Serial.print(fc5);Serial.print(" ");
       Serial.print(fc6);Serial.print(" ");Serial.print(fc7);Serial.print(" ");Serial.print(fc8);Serial.print("
");Serial.print(fc9);Serial.print(" ");Serial.println(fc10);
       Serial.println("No. of absent: ");
       Serial.println(ab);
        Serial.println("No. of attended: ");
       Serial.println(at);
       Serial.println("No. of arrived: ");
       Serial.println(ar); //testing
       if( at> ar && arvi==1){
         Serial.println("Warning not all students have left the bus");
       }
//theend:
  Serial.println("Out");

}
```

3) Attendance code for passive RFID to Arduino mega:

```
#include <GSM.h>
#define PINNUMBER ""
GSM gsmAccess;
GSM_SMS sms;
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <MFRC522.h> // for the RFID
#include <SPI.h> // for the RFID and SD card module
#include <SD.h> // for the SD card
#include <RTClib.h> // for the RTC
// define pins for RFID
#define CS_RFID 53
#define RST_RFID 5
// define select pin for SD card module
#define CS_SD 4
// Create a file to store the data
File myFile;
// Instance of the class for RFID
MFRC522 rfid(CS_RFID, RST_RFID);
// Variable to hold the tag's UID
String Student;
// Instance of the class for RTC
RTC_DS1307 rtc;
```

```
// Define check in time
const int checkInHour = 01;
const int checkInMinute = 00;
const int classwithadd= checkInMinute + 5 ;
const int latepolicy= checkInHour +15;
//Variable to hold user check in
int userCheckInHour;
int userCheckInMinute;

int TimeHour;
int TimeMinute;
int zmessage;
int ab=5;
int x;
int i;
String y;
 String Students[6]={"Ahmed", "Khalid", "Omar", "Fahad", "Saleh","          "};
// Pins for LEDs and buzzer
const int redLED = 8;
const int greenLED = 6;
const int buzzer = 23;
void setup() {
  lcd.init();                    // initialize the lcd
  lcd.backlight();
  // Set LEDs and buzzer as outputs
  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(buzzer, OUTPUT);
  // Init Serial port
  Serial.begin(9600);
  while(!Serial); // for Leonardo/Micro/Zero
  Serial.println("SMS Messages Sender");
 // connection state
  boolean notConnected = true;

  // Start GSM shield
  // If your SIM has PIN, pass it as a parameter of begin() in quotes
  while (notConnected) {
    if (gsmAccess.begin(PINNUMBER) == GSM_READY) {
      notConnected = false;
    } else {
      Serial.println("Not connected");
      lcd.print("GSM not Connected");
      delay(1000);
    }
  }
  Serial.println("GSM initialized");
  lcd.print("GSM Ready");
  delay(1000);
    lcd.clear();
  // Init SPI bus
  SPI.begin();
  // Init MFRC522
  rfid.PCD_Init();

  // Setup for the SD card
  Serial.print("Initializing SD card...");
  lcd.print("Initializing ");
  lcd.setCursor(0, 1);
```

```
  lcd.print("SD card...");
  delay(3000);
  lcd.clear();
  if(!SD.begin(CS_SD)) {
    Serial.println("initialization failed!");
    lcd.print("Initializing ");
    lcd.setCursor(0, 1);
    lcd.print("failed!");
    return;
  }
  Serial.println("initialization done.");
  Serial.println("SMS Messages Sender");
  delay(200);
lcd.print("Please Swipe ID ");
  // Setup for the RTC
  if(!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    lcd.clear();
    lcd.print("Couldn't find RTC");
    while(1);
  }
  else {
    // following line sets the RTC to the date & time this sketch was compiled
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
  if(!rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    lcd.clear();
    lcd.print("RTC Not Running!");
  }
 }
 void loop() {
  //look for new cards
   MFRC522::MIFARE_Key key;
   for (byte i = 0; i < 6; i++) {
            key.keyByte[i] = 0xFF;
      }
      // Look for new cards
      if ( ! rfid.PICC_IsNewCardPresent()) {
            return;
      }

      // Select one of the cards
      if ( ! rfid.PICC_ReadCardSerial()) {
            return;
      }
    readRFID();
   absentsSMS();
  delay(10);
 }

 void readRFID() {
   lcd.clear();
  Serial.print("Card UID:");
      for (byte i = 0; i < rfid.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 = rfid.PICC_GetType(rfid.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 59 98 E3 2A
        if( (rfid.uid.uidByte[0] == 0x59) && (rfid.uid.uidByte[1] == 0x98) && (rfid.uid.uidByte[2] == 0xE3)
&& (rfid.uid.uidByte[3] == 0x2A) ) // student1 card
        {
          tone(buzzer, 2000);
  delay(200);
  noTone(buzzer);
 ab= ab-1;
  delay(200);
          lcd.clear();
          lcd.print("Attended");
          delay(1000);
          lcd.clear();
          lcd.print("Welcome Ahmed");
         Serial.println("Ahmed");
          delay(1000);
       Student= "Ahmed";
         // for gsm

            logCard();
      verifyCheckIn();
          delay(1000);
        lcd.clear();
         lcd.print("Message sent");
         DateTime now = rtc.now();
         sms.beginSMS("+966544400207");
  sms.print("Ahmed Attended At ");
   sms.print(now.year(), DEC);
    sms.print('/');
    sms.print(now.month(), DEC);
    sms.print('/');
    sms.print(now.day(), DEC);
    sms.print(',');
    sms.print(now.hour(), DEC);
    sms.print(':');
    sms.println(now.minute(), DEC);
 for(i=1; i<=5; i++){ // find student and make him absent
if(Student == Students[i]){
y=Students[i];
Students[i]=Students[ab];
Students[ab]=y;
break;
}
   }
  sms.endSMS();
        delay(2000);
                lcd.clear();
          lcd.print("Swipe Your ID");
       return;
```

```
      }
                  // student29 A2 14 A4
      if( (rfid.uid.uidByte[0] == 0x29) && (rfid.uid.uidByte[1] == 0xA2) && (rfid.uid.uidByte[2] == 0x14)
&& (rfid.uid.uidByte[3] == 0xA4) ) // student2 card
      {
          tone(buzzer, 2000);
  delay(200);
  noTone(buzzer);
  ab= ab-1;
       lcd.clear();
        lcd.print("Attended");
        delay(1000);
        lcd.clear();
        lcd.print("Welcome Khaled");
       Serial.println("Khaled");
        delay(1000);
     Student= "Khaled";
       // for gsm
       logCard();
    verifyCheckIn();
      delay(1000);
      lcd.clear();
       lcd.print("Message sent");
       DateTime now = rtc.now();
        sms.beginSMS("+966544400207");
  sms.print("Khaled Attended At ");
   sms.print(now.year(), DEC);
    sms.print('/');
    sms.print(now.month(), DEC);
    sms.print('/');
    sms.print(now.day(), DEC);
    sms.print(',');
    sms.print(now.hour(), DEC);
    sms.print(':');
    sms.println(now.minute(), DEC);
   for(i=1; i<=5; i++){ // find student and make him absent
if(Student == Students[i]){
y=Students[i];
Students[i]=Students[ab];
Students[ab]=y;
break;
}
   }
  sms.endSMS();
       delay(1000);
             lcd.clear();
        lcd.print("Swipe Your ID");

       return;
      }
      if( (rfid.uid.uidByte[0] == 0xE9) && (rfid.uid.uidByte[1] == 0x56) && (rfid.uid.uidByte[2] == 0x57)
&& (rfid.uid.uidByte[3] == 0xA2) ) // student1 card
      {
            tone(buzzer, 2000);
  delay(200);
  noTone(buzzer);
  ab= ab-1;
       lcd.clear();
       lcd.print("Attended");
       delay(1000);
```

```
          lcd.clear();
          lcd.print("Welcome Omar");
         Serial.println("Omar");
          delay(1000);
       Student= "Omar";
         // for gsm
        logCard();
      verifyCheckIn();
          delay(1000);
        lcd.clear();
         lcd.print("Message sent");
          sms.beginSMS("+966544400207");
          DateTime now = rtc.now();
  sms.print("Omar Attended At ");
  sms.print(now.year(), DEC);
    sms.print('/');
    sms.print(now.month(), DEC);
    sms.print('/');
    sms.print(now.day(), DEC);
    sms.print(',');
    sms.print(now.hour(), DEC);
    sms.print(':');
    sms.println(now.minute(), DEC);
  sms.endSMS();
    for(i=1; i<=5; i++){ // find student and make him absent
if(Student == Students[i]){
y=Students[i];
Students[i]=Students[ab];
Students[ab]=y;
break;
}
   }
         delay(1000);
         lcd.clear();
          lcd.print("Swipe Your ID");
      return;
        }
        if( (rfid.uid.uidByte[0] == 0x69) && (rfid.uid.uidByte[1] == 0x5A) && (rfid.uid.uidByte[2] == 0x04)
&& (rfid.uid.uidByte[3] == 0xA4) ) // student1 card
        {
               tone(buzzer, 2000);
  delay(200);
  noTone(buzzer);
  ab= ab-1;
          lcd.clear();
          lcd.print("Attended");
          delay(1000);
          lcd.clear();
          lcd.print("Welcome Fahad");
         Serial.println("Fahad");
          delay(1000);
       Student= "Fahad";
         // for gsm
           logCard();
      verifyCheckIn();
          delay(1000);
        lcd.clear();
         lcd.print("Message sent");
         DateTime now = rtc.now();
            sms.beginSMS("+966544400207");
```

```
  sms.print("Fahad Attended At");
   sms.print(now.year(), DEC);
    sms.print('/');
    sms.print(now.month(), DEC);
    sms.print('/');
    sms.print(now.day(), DEC);
    sms.print(',');
    sms.print(now.hour(), DEC);
    sms.print(':');
    sms.println(now.minute(), DEC);
  sms.endSMS();
    for(i=1; i<=5; i++){ // find student and make him absent
if(Student == Students[i]){
y=Students[i];
Students[i]=Students[ab];
Students[ab]=y;
break;
}
   }
        delay(1000);

        lcd.clear();
         lcd.print("Swipe Your ID");
      return;
       }
       if( (rfid.uid.uidByte[0] == 0xD9) && (rfid.uid.uidByte[1] == 0x30) && (rfid.uid.uidByte[2] == 0x04)
&& (rfid.uid.uidByte[3] == 0xA4) ) // student1 card
       {
              tone(buzzer, 2000);
  delay(200);
  noTone(buzzer);
  ab= ab-1;
        lcd.clear();
        lcd.print("Attended");
        delay(1000);
        lcd.clear();
        lcd.print("Welcome Saleh");

        Serial.println("Saleh");
         delay(1000);
      Student= "Saleh";
       // for gsm
                logCard();
     verifyCheckIn();
        delay(1000);
       lcd.clear();
        lcd.print("Message sent");
        DateTime now = rtc.now();
        sms.beginSMS("+966544400207");
  sms.print("Saleh Attended At");
   sms.print(now.year(), DEC);
    sms.print('/');
    sms.print(now.month(), DEC);
    sms.print('/');
    sms.print(now.day(), DEC);
    sms.print(',');
    sms.print(now.hour(), DEC);
    sms.print(':');
    sms.println(now.minute(), DEC);
     for(i=1; i<=5; i++){ // find student and make him absent
```

```
if(Student == Students[i]){
y=Students[i];
Students[i]=Students[ab];
Students[ab]=y;
break;
}
  }
 sms.endSMS();

       delay(2000);

       lcd.clear();
        lcd.print("Swipe Your ID");
     return;
       }

      else

      Serial.println("unregistered user");


 tone(buzzer, 2000);
 delay(200);
 noTone(buzzer);
 lcd.print("Wrong class");
 delay(2000);
 lcd.clear();
 lcd.print("Swipe your ID");
 // Sound the buzzer when a card is read
 delay(200);
}
void logCard() {
 // Enables SD card chip select pin
 digitalWrite(CS_SD,LOW);
  // Open file
 myFile=SD.open("DATA.txt", FILE_WRITE);
 // If the file opened ok, write to it
 if (myFile) {
   Serial.println("File opened ok");
   lcd.clear();
   lcd.print("File opened ok");
   delay(2000);
   myFile.print(Student);
   myFile.print(" , ");
   // Save time on SD card
   DateTime now = rtc.now();
   myFile.print(now.year(), DEC);
   myFile.print('/');
   myFile.print(now.month(), DEC);
   myFile.print('/');
   myFile.print(now.day(), DEC);
   myFile.print(',');
   myFile.print(now.hour(), DEC);
   myFile.print(':');
   myFile.println(now.minute(), DEC);
   // Print time on Serial monitor
   Serial.print(now.year(), DEC);
   Serial.print('/');
   Serial.print(now.month(), DEC);
   Serial.print('/');
```

```
    Serial.print(now.day(), DEC);
    Serial.print(' ');
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.println(now.minute(), DEC);
    Serial.println("sucessfully written on SD card");
    lcd.clear();
    lcd.print(now.year(), DEC);
    lcd.print(':');
    lcd.print(now.month(), DEC);
    lcd.print(':');
    lcd.print(now.day(), DEC);
    lcd.print(' ');
    lcd.setCursor(11, 0);
    lcd.print(now.hour(), DEC);
    lcd.print(':');
    lcd.print(now.minute(), DEC);
    lcd.setCursor(0, 1);
    lcd.print("Written on SD...");
    delay(2000);
    myFile.close();
    // Save check in time;
    userCheckInHour = now.hour();
    userCheckInMinute = now.minute();
  }
  else {
    Serial.println("error opening data.txt");
    lcd.clear();
    lcd.print("error opening data.txt");
  }
  // Disables SD card chip select pin
  digitalWrite(CS_SD,HIGH);
}

void verifyCheckIn(){
  if((userCheckInHour < checkInHour)||((userCheckInHour==checkInHour) && (userCheckInMinute <=
checkInMinute))){
    digitalWrite(greenLED, HIGH);
    delay(2000);
    digitalWrite(greenLED,LOW);
    Serial.println("On time");
    lcd.clear();
    lcd.print("On time");
  }
   else  if ((userCheckInHour == checkInHour) && ( userCheckInMinute > (classwithadd)) && (
userCheckInMinute <= (latepolicy) ) ){ //// 16 min for late
    digitalWrite(redLED, HIGH);
    delay(2000);
    digitalWrite(redLED,LOW);
    Serial.println("You are late...");
    lcd.clear();
    lcd.print("You are Late...");
    delay(3000);
    lcd.clear();
  }
  else{
      Serial.println ("did not attend" );
  }
}
void absentsSMS(){
```

```
  DateTime now = rtc.now();
 TimeHour = now.hour();//keep checking time
   TimeMinute = now.minute(); //keep checking time
 if ((TimeHour==7) && (TimeMinute==30) && (zmessage==0)){ //at7:30 start sending sms
switch(ab){
 case 1:
 for(x=1; x<=1; x++){
   sms.print(Students[x]);
   sms.print("is absent from school at ");
  sms.print(now.year(), DEC);
   sms.print('/');
   sms.print(now.month(), DEC);
   sms.print('/');
   sms.print(now.day(), DEC);
   sms.print(',');
   sms.print(now.hour(), DEC);
   sms.print(':');
   sms.println(now.minute(), DEC);
 sms.endSMS();
 }
 break;

 case 2:
 for(x=1; x<=2; x++){
   sms.print(Students[x]);
   sms.print("is absent from school at ");
  sms.print(now.year(), DEC);
   sms.print('/');
   sms.print(now.month(), DEC);
   sms.print('/');
   sms.print(now.day(), DEC);
   sms.print(',');
   sms.print(now.hour(), DEC);
   sms.print(':');
   sms.println(now.minute(), DEC);
 sms.endSMS();
 }
 break;

 case 3:
 for(x=1; x<=3; x++){
   sms.print(Students[x]);
   sms.print("is absent from school at ");
  sms.print(now.year(), DEC);
   sms.print('/');
   sms.print(now.month(), DEC);
   sms.print('/');
   sms.print(now.day(), DEC);
   sms.print(',');
   sms.print(now.hour(), DEC);
   sms.print(':');
   sms.println(now.minute(), DEC);
 sms.endSMS();
 }
 break;
 case 4:
 for(x=1; x<=4; x++){
   sms.print(Students[x]);
   sms.print("is absent from school at ");
  sms.print(now.year(), DEC);
```

```
  sms.print('/');
  sms.print(now.month(), DEC);
  sms.print('/');
  sms.print(now.day(), DEC);
  sms.print(',');
  sms.print(now.hour(), DEC);
  sms.print(':');
  sms.println(now.minute(), DEC);
 sms.endSMS();
 }
 break;

 case 5:
 for(x=1; x<=5; x++){
  sms.print(Students[x]);
  sms.print("is absent from school at ");
 sms.print(now.year(), DEC);
  sms.print('/');
  sms.print(now.month(), DEC);
  sms.print('/');
  sms.print(now.day(), DEC);
  sms.print(',');
  sms.print(now.hour(), DEC);
  sms.print(':');
  sms.println(now.minute(), DEC);
 sms.endSMS();
 }
 break;
 }
 zmessage=1;//to send sms once
}
}
```

4) Blynk code to find the location using Google map and measure the speed

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
WidgetMap myMap(V1);

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "3canLYJButK9aoKkoms_pw84BST1t6WN";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Bahha";
char pass[] = "01234560";

WidgetMap myMap(V1);
#include <DFRobot_sim808.h>
#include <SoftwareSerial.h>

//#define PIN_TX    10
//#define PIN_RX    11
//SoftwareSerial mySerial(PIN_TX,PIN_RX);
//DFRobot_SIM808 sim808(&mySerial);//Connect RX,TX,PWR,

DFRobot_SIM808 sim808(&Serial);
```

```
void setup() {
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

  // If you want to remove all points:
  //myMap.clear();

  int index = 0;
  float lat = 51.5074;
  float lon = 0.1278;
  myMap.location(index, lat, lon, "value");
  //mySerial.begin(9600);
  Serial.begin(9600);

  //******** Initialize sim808 module *************
  while(!sim808.init()) {
     delay(1000);
     Serial.print("Sim808 init error\r\n");
  }

  //************* Turn on the GPS power************
  if( sim808.attachGPS())
     Serial.println("Open the GPS power success");
  else
     Serial.println("Open the GPS power failure");

}

void loop() {
  Blynk.run();
   //*************** Get GPS data *******************
   if (sim808.getGPS()) {
     Serial.print(sim808.GPSdata.year);
     Serial.print("/");
     Serial.print(sim808.GPSdata.month);
     Serial.print("/");
     Serial.print(sim808.GPSdata.day);
     Serial.print(" ");
     Serial.print(sim808.GPSdata.hour);
     Serial.print(":");
     Serial.print(sim808.GPSdata.minute);
     Serial.print(":");
     Serial.print(sim808.GPSdata.second);
     Serial.print(":");
     Serial.println(sim808.GPSdata.centisecond);

     Serial.print("latitude :");
     Serial.println(sim808.GPSdata.lat,6);

     sim808.latitudeConverToDMS();
     Serial.print("latitude :");
     Serial.print(sim808.latDMS.degrees);
     Serial.print("\^");
     Serial.print(sim808.latDMS.minutes);
     Serial.print("\'");
     Serial.print(sim808.latDMS.seconeds,6);
```

```
    Serial.println("\"");
    Serial.print("longitude :");
    Serial.println(sim808.GPSdata.lon,6);
    sim808.LongitudeConverToDMS();
    Serial.print("longitude :");
    Serial.print(sim808.longDMS.degrees);
    Serial.print("\^");
    Serial.print(sim808.longDMS.minutes);
    Serial.print("\'");
    Serial.print(sim808.longDMS.seconeds,6);
    Serial.println("\"");

    Serial.print("speed_kph :");
    Serial.println(sim808.GPSdata.speed_kph);
    Serial.print("heading :");
    Serial.println(sim808.GPSdata.heading);

    //************* Turn off the GPS power ************
    sim808.detachGPS();
  }

}
```