International Journal of Computer Techniques – Volume 12 Issue 4,
July - August - 2025

Smart Attendance System

VAIBHAV BORKAR PRAVEEN VISHWAKARMA LOCHAN YADAV VAIBHAV

Dr Arif Hasan
arifhasanbusiness@gmail.com
(Guide)

Abstract

The RFID-Based Attendance System is an innovative solution designed to automate the process of recording and managing attendance in educational institutions and organizations. Traditional attendance systems are time-consuming, prone to errors, and vulnerable to proxy attendance. This project aims to overcome these challenges by using RFID (Radio Frequency Identification) technology integrated with a backend database system. In this project, each student or employee is assigned a unique RFID card, which they can scan using an RFID reader connected to a microcontroller (such as Raspberry Pi or Arduino). A Python script running on the system reads the RFID data in real time and sends it to a Spring Boot-based REST API. The backend processes the data and stores it securely in a MySQL database. The system ensures accurate, fast, and reliable attendance logging. It also allows administrators to view attendance records, manage users, and prevent duplicate entries. The integration of hardware and software provides a complete solution for smart attendance management. This project demonstrates the practical application of embedded systems, web services, and databases, offering a cost-effective and scalable alternative to manual attendance systems.

1. Introduction

This project presents a cost-effective, IoT-enabled RFID-based attendance system designed to replace traditional manual methods in educational and professional settings. The system uses RFID cards assigned to individuals, which, when scanned via an RC522 reader connected to an Arduino UNO, transmit a unique ID and timestamp to a Python script. This script forwards the data to a Spring Boot REST API for validation and storage in a MySQL database.

The system ensures real-time attendance tracking, prevents proxy marking, and significantly reduces manual effort and errors. Admins can access and analyze attendance records through a web interface or database tools. The solution is scalable and can be extended with features like dashboards, notifications, biometrics, or cloud integration. This project integrates embedded systems, backend development, and database management to deliver a practical and efficient automation tool for institutional use.

2. Literature Review

Table 1: Comparison of Recent RFID/IoT-based Smart Attendance Projects :

S.No	Title	Technology Used	Cloud/Backend	Hardware/Sensors	Drawbacks
1	RFID-Based Student Attendance System (2019)	Arduino UNO + RFID RC522	Local Database	RFID Reader, Cards	No remote access, manual report gen.
2	IoT-Enabled Smart Attendance System (2020)	Arduino + ESP8266	Firebase	RFID Reader	Wi-Fi only, limited offline support
3	RFID and GSM-Based Attendance Management (2021)	Arduino + GSM	Custom Server	RFID Reader	Higher cost due to GSM charges
4	Face Recognition & RFID Hybrid Attendance (2022)	Raspberry Pi + Camera + RFID	AWS IoT	Camera, RFID	High processing & cost
5	Cloud-Based Attendance System using RFID (2022)	ESP32 + RFID + Python	Blynk	RFID Reader	Needs constant internet connection
6	Long-Range UHF RFID Attendance Tracking (2023)	STM32 + UHF RFID	Azure IoT Hub	UHF RFID Antenna	Expensive hardware setup
7	Solar-Powered IoT RFID Attendance Node (2023)	ESP32 + Solar + RFID	Google Cloud	RFID, Solar Module	High cost due to solar integration

3. Methodology

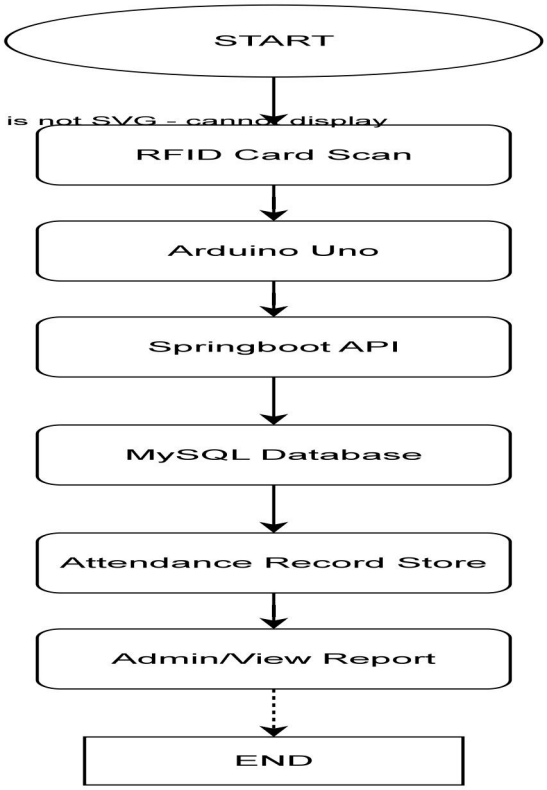
3.1 Components Used

- -Microcontroller: Arduino UNO
- RFID Reader Module: RC522
- RFID Cards/Tags: Unique UID cards assigned to each user
- Jumper Wires & Breadboard: For circuit connections and testing
- USB Cable: Power supply and serial communication with PC
- Computer System: Runs Python middleware and backend communication
- Power Supply: 5V (via USB or adapter)

3.2 Circuit Diagram

Figure 1 below shows a sample circuit diagram for the Smart Attendance System

Figure 1: Circuit Diagram



3.3 PCB Layout and Admin Interface

Figure 2 and Figure 3 will illustrate the PCB design and Admin dashboard respectively (placeholders).

Figure 2: PCB Layout

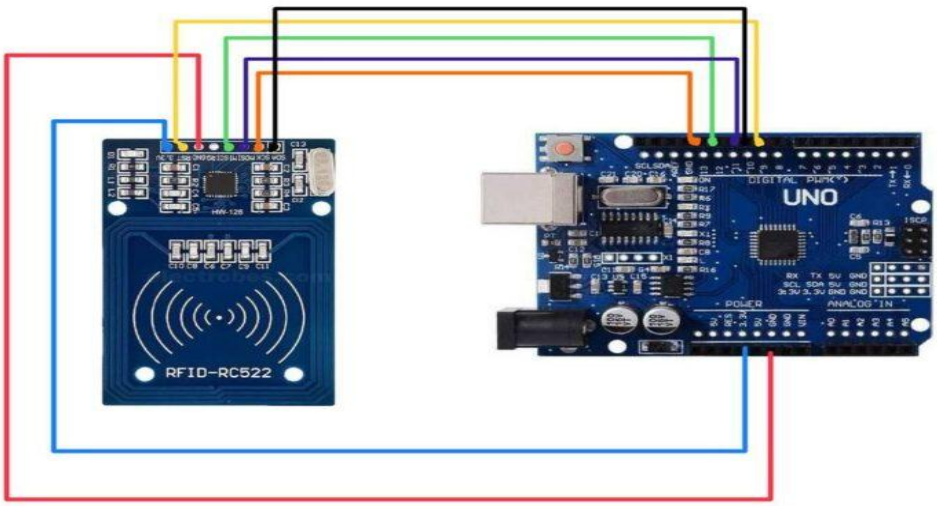
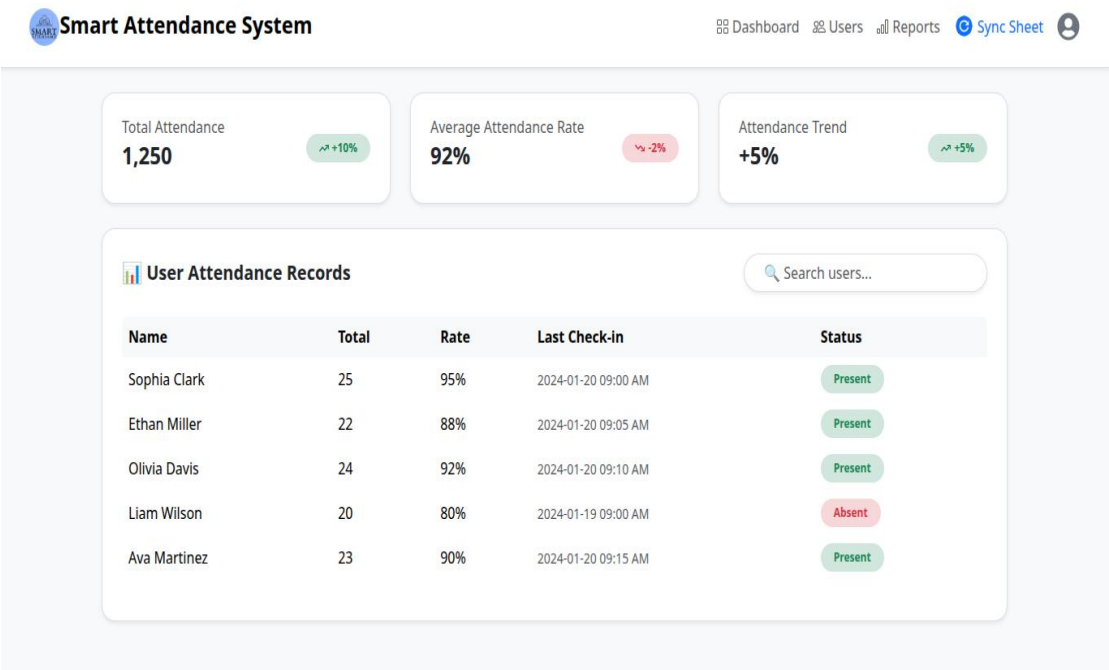


Figure 3: Admin Dashboard



4. Sample Arduino Code

```
#include <SPI.h>

#include <MFRC522.h>

#define RESET_PIN 9

#define SELECT_PIN 10

MFRC522 rfidReader(SELECT_PIN, RESET_PIN);

int serialCounter = 1;

unsigned long startMillis;

int startHours = 12, startMinutes = 55, startSeconds = 50;

struct UserData {

  String cardUID;

  String rollNumber;

  String name;

  String lastTimeIn;

  bool isInside;
```

```
};
UserData userList[] = {
  {"46e10a2957980", "HST-01", "Harsh Vardhan", "", false},
  {"6a9df0", "HST-02", "Viraj", "", false},
  {"5f93979", "HST-03", "Vaayu", "", false},
};
const int userCount = sizeof(userList) / sizeof(userList[0]);
String getCurrentTime() {
  unsigned long elapsedSeconds = (millis() - startMillis) / 1000;
  int currentHours = startHours + (elapsedSeconds / 3600);
  int currentMinutes = startMinutes + ((elapsedSeconds % 3600) / 60);
  int currentSeconds = startSeconds + (elapsedSeconds % 60);
  if (currentSeconds >= 60) {
    currentMinutes += currentSeconds / 60;
    currentSeconds %= 60;
  }
  if (currentMinutes >= 60) {
    currentHours += currentMinutes / 60;
    currentMinutes %= 60;
  }
  if (currentHours >= 24) {
    currentHours %= 24;
  }
  char formattedTime[9];
  sprintf(formattedTime, "%02d:%02d:%02d", currentHours, currentMinutes, currentSeconds);
  return String(formattedTime);
}
int findUserIndex(String cardNumber) {
```

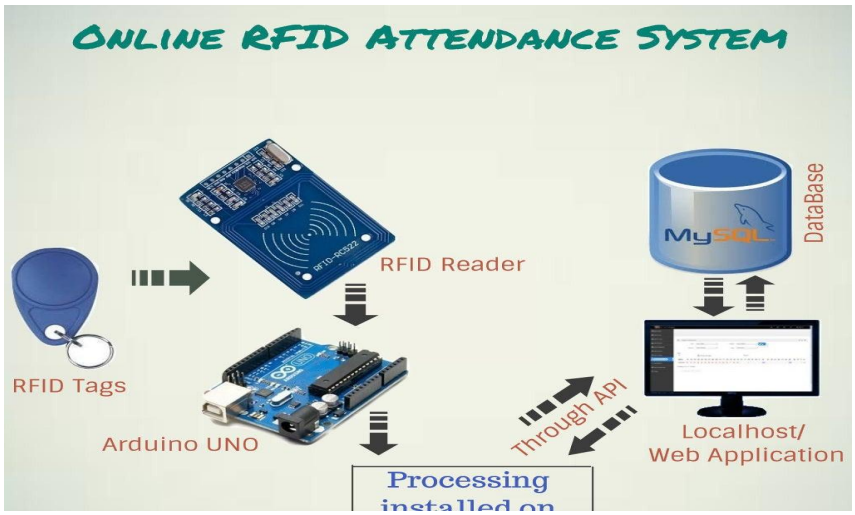
```
 for (int i = 0; i < userCount; i++) {
  if (userList[i].cardUID == cardNumber) {
   return i;
  }
 }
 return -1;
}
void setup() {
 Serial.begin(9600);
 SPI.begin();
 rfidReader.PCD_Init();
 Serial.println("Serial Number,RFID Card Number,Roll Number,Name,Time In,Time Out,Status");
}
void loop() {
 if (!rfidReader.PICC_IsNewCardPresent() || !rfidReader.PICC_ReadCardSerial()) {
  return;
 }
 String cardNumber = "";
 for (byte i = 0; i < rfidReader.uid.size; i++) {
  cardNumber += String(rfidReader.uid.uidByte[i], HEX);
 }
 String currentTime = getCurrentTime();
 String timeIn = "-", timeOut = "-", status = "Unknown";
 int userIndex = findUserIndex(cardNumber);
 if (userIndex != -1) {
  UserData &user = userList[userIndex];
  if (!user.isInside) {
   user.lastTimeIn = currentTime;
```

```
    user.isInside = true;
    timeIn = user.lastTimeIn;
    status = "STUDENT ENTERED";
  } else {
    timeIn = user.lastTimeIn;
    timeOut = currentTime;
    user.isInside = false;
    status = "STUDENT EXITED";
  }
  Serial.print(serialCounter++);
  Serial.print(",");
  Serial.print(cardNumber);
  Serial.print(",");
  Serial.print(user.rollNumber);
  Serial.print(",");
  Serial.print(user.name);
  Serial.print(",");
  Serial.print(timeIn);
  Serial.print(",");
  Serial.print(timeOut);
  Serial.print(",");
  Serial.println(status);
 } else {
  Serial.print(serialCounter++);
  Serial.print(",");
  Serial.print(cardNumber);
  Serial.print(",");
  Serial.print("Unknown");
```

```
    Serial.print(",");
    Serial.print("Unknown");
    Serial.print(",");
    Serial.print(currentTime);
    Serial.print(",");
    Serial.print("-");
    Serial.print(",");
    Serial.println("STUDENT ENTERED");
  }
  delay(1000);
  rfidReader.PICC_HaltA();
  rfidReader.PCD_StopCrypto1();
}
```

5. Sample Pictures

Figure 4: Assembled Prototype



6. Conclusion

The proposed Smart Attendance System using RFID and IoT effectively overcomes the limitations of manual attendance methods by providing an automated, accurate, and time-efficient solution. With its low-cost hardware, scalable architecture, and capability to integrate with cloud platforms, the system ensures real-time tracking and secure data storage. Its modular design makes it suitable for schools, colleges, and corporate organizations,

improving transparency and reducing errors. Comparative analysis with existing systems demonstrates its advantages in affordability, reliability, and ease of deployment.

7. References

- MFRC522 RFID Module Datasheet – NXP Semiconductors.
- Arduino Uno Technical Reference – Arduino.cc.
- ESP8266 Wi-Fi Module Documentation – Espressif Systems.
- Klaus Finkenzeller, *RFID Handbook: Fundamentals and Applications* (3rd Edition), Wiley.
- IEEE Paper – "IoT-Based Smart Attendance Monitoring System," 2022.
- MySQL Documentation – https://dev.mysql.com/doc/
- Python Serial Communication Library (pyserial) Documentation – https://pyserial.readthedocs.io

1.