# Smart Door Lock System Using RFID and Blynk

## INTRODUCTION

In today's world, security and automation play a crucial role in both residential and commercial spaces. This project presents a Smart Door Lock System that combines RFID technology and IoT control using Blynk to enhance access control. It allows a door to be unlocked either by scanning a known RFID tag or remotely via a smartphone application.

## PROBLEM STATEMENT

In today's fast-paced world, ensuring home and office security has become a growing concern. Traditional lock and key systems, though widely used, present several vulnerabilities:

**1.Risk of Lost or Stolen Keys:** Physical keys can be easily lost, stolen, or duplicated, leading to unauthorized access.

**2.Lack of Access Records**: There is no way to track who accessed the door and when.

**3.Inconvenience of Manual Operation:** Physically unlocking and locking doors is inconvenient, especially when managing access for multiple people.

**4.Limited Remote Control**: Traditional locks offer no way to remotely control or monitor access, which is a necessity in smart environments.

Given these challenges, the need for a smart, secure, and remotely controllable door access system became evident. The integration of RFID authentication with Blynk IoT platform offers a scalable and cost-effective solution to:

1.Allow only authorized persons to unlock doors via RFID tags.

2.Log access events with real-time timestamps.

3.Enable users to remotely unlock/lock the door using a smartphone.

4.Improve safety through automated responses to unauthorized access attempts.

This project aims to address these problems with a simple yet robust smart door locking system built on affordable and easily programmable hardware like the ESP8266.

## PURPOSE AND USE OF THE PROJECT

**Secure and Contactless Access with RFID Tags:**

RFID technology provides a contactless method for identification and authentication. Each RFID tag carries a unique identifier that can be read by an RFID reader. When the user holds their RFID tag near the reader, the system verifies the tag's authenticity, granting or denying access. This eliminates the need for physical keys, reducing wear and tear and the risk of key duplication. It also enhances security by ensuring only authorized users can access the premises.

**Remote Control via the Blynk Mobile Application:**

The Blynk app acts as a remote control interface for your IoT system. It allows users to lock or unlock the door from anywhere in the world, as long as they have an internet connection. This is especially useful for situations where physical access to the location is not possible, such as when granting entry to visitors or family members remotely. The app can be customized to provide various functionalities, like viewing the status of the door, receiving notifications, and even logging access events.

**Monitoring Door Access with Timestamps:**

To maintain a secure and auditable record of all access events, the system logs each door access attempt along with a timestamp. This feature is important for keeping track of who accessed the premises and when, helping to identify unauthorized access attempts or simply keeping a record for security audits. The access logs can be accessed through the mobile application or a web interface, providing real-time monitoring and historical data.

**Improvement Over Traditional Locking Systems:**

Traditional locking systems often rely on physical keys or manual entry, which can be lost, stolen, or copied. Your IoT-based access system eliminates these issues by offering a digital, secure, and remote-controlled solution. It also improves the user experience by providing more flexible and advanced features, such as time-based access, remote control, and real-time monitoring. Additionally, the cost-effectiveness and ease of integration with existing infrastructure make this system an attractive alternative for homes, offices, hostels, and labs.

The overall goal of this system is to provide a low-cost yet highly secure and convenient solution for access control, making it suitable for a variety of environments where controlled and safe access is a priority.

## COMPONENTS REQUIRED

| Component | Description |
|---|---|
| ESP8266 (NodeMCU) | Microcontroller with built-in Wi-Fi |
| MFRC522 Module | RFID reader to detect RFID cards |
| RFID Tag/Card | Used to identify authorized users |
| Relay Module | To control the solenoid lock |
| Solenoid Lock | Electronic door lock |
| Buzzer | Sound alert for unauthorized access |
| LED | Visual indicator for access status |
| Blynk App | Mobile app to control and monitor access |
| Internet/Wi-Fi | For connecting ESP8266 to Blynk server |
| Power Supply | 5V/9V supply for the circuit |
| Jumper Wires, Breadboard | For circuit assembly |

## Project Overview

### RFID Card Scan for Secure Entry:

Known RFID Tags: When an authorized user scans their RFID card, the system recognizes the tag and unlocks the door for a brief period. This action provides secure entry, ensuring that only authorized individuals can gain access.

Timestamp and Notification: After the door unlocks, the system sends a message with the timestamp of the event to the Blynk app. This ensures that the user (or system administrator) is notified of the event in real-time, which can be useful for tracking and auditing purposes.

**Handling Unknown RFID Tags:**

If the system detects an RFID tag that is not recognized or is not authorized, it triggers a buzzer to alert those nearby of an attempted unauthorized entry. Additionally, an alert message is sent to the Blynk app, which notifies the owner or administrator of the access attempt, adding an extra layer of security.

**Remote Control via the Blynk App:**

Virtual Switch for Manual Lock/Unlock: The Blynk app also provides a virtual switch that allows users to lock or unlock the door manually. This feature can be used from anywhere in the world, as long as there's an internet connection, making it ideal for situations where someone needs remote access to the premises (e.g., granting entry to visitors or remote staff). The ability to control the door remotely ensures flexibility and convenience in various scenarios.

## FLOW OF THE PROJECT

1. **System Initialization:**
   - Wi-Fi is connected.
   - RFID reader and pins are initialized.
   - Time is configured using NTP.

2. **Card Scanning:**
   - System constantly checks for a new RFID tag.
   - If a card is present, the UID is read.

3. **UID Comparison:**
   - The scanned UID is compared with a predefined UID.
   - If it matches:
     - Unlocks door via relay.
     - Activates LED.
     - Sends access log to Blynk.
   - If not:
     - Keeps door locked.
     - Activates buzzer.
     - Sends denial alert to Blynk.

4. **Remote Unlock:**
   - A Blynk virtual button allows unlocking from the mobile app.
   - Status and timestamp are updated in Blynk.

## COMPONENT EXPLANATION

### 1. ESP8266 NodeMCU

Function: Acts as the brain of the entire system.

Role: Connects to a Wi-Fi network and communicates with the Blynk server to receive/send commands and data.

Operations: Handles all logic, processes RFID inputs, controls the relay, buzzer, and LEDs based on conditions, and sends logs to the Blynk app.

### 2. MFRC522 RFID Module

Function: Reads the UID (Unique Identifier) from RFID tags/cards.

Communication: Uses SPI (Serial Peripheral Interface) to communicate with the ESP8266.

Purpose: Identifies whether the scanned tag is authorized or unauthorized

### 3. Relay and Solenoid Lock

**Relay:**

An electrically controlled switch.

Function: Turns the solenoid lock ON or OFF based on instructions from the ESP8266.

**Solenoid Lock:**

An electromagnetic locking mechanism.

Function: Remains locked by default and unlocks briefly when the relay is activated.

**4. Buzzer and LED**

Buzzer:

Function: Emits a sound to signal unauthorized access or system alerts.

LED:

Function: Visual indicator of the lock status (e.g., green when unlocked, red when locked)

**5. Blynk Application**

Function: Mobile-based control interface.

Features:

Remote locking/unlocking using a virtual button.

Sends notifications and access logs with timestamps.

Displays real-time system status.

Platform: Cloud-connected via Blynk server and ESP8266.

**Classification: Sensors vs Actuators**

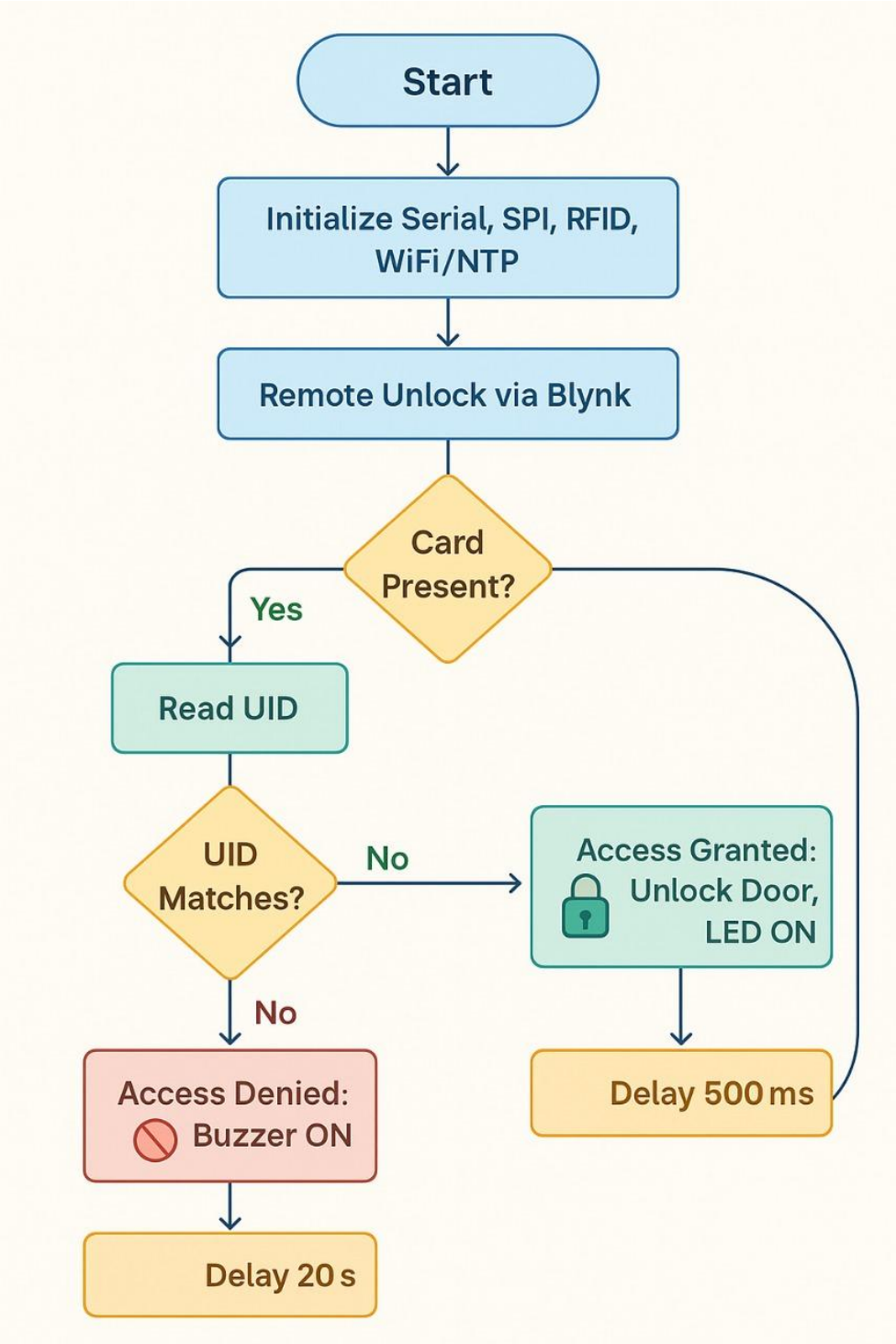
**Sensors (Detect and send data):**

| **Component** | **Role** |
|---|---|
| MFRC522 RFID- | Detects the presence and identity of RFID tags/cards. |

**Actuators (Act based on received signals):**

| **Component** | **Role** |
|---|---|
| Relay Module- | Controls electrical power to the solenoid lock. |
| Solenoid Lock- | Unlocks/locks the door physically. |
| Buzzer- | Produces sound alerts for events (e.g., intruder). |
| LED | Displays visual status (locked/unlocked). |

## FLOW DIAGRAM:



## PICTORIAL REPRESENTATION:

**SOURCE CODE:**

```
#define BLYNK_TEMPLATE_ID "TMPL3qvrx_wEH"

#define BLYNK_TEMPLATE_NAME "openTheDoor"

#define BLYNK_AUTH_TOKEN "EhVmO9uSlbOrSAZ4a5c5WhOYt5f_WFPI"

#include <SPI.h>

#include <MFRC522.h>

#include <time.h>  // For getting time via NTP


#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>


// WiFi Credentials

char ssid[] = "Vivo v40";

char pass[] = "abinayat092";


// Pin Definitions

#define SS_PIN D2         // RFID SDA

#define RST_PIN D4        // RFID RST

#define LOCK_PIN D0       // Relay for solenoid

#define led D1            // Access LED

#define buz D8            // Buzzer for denied


MFRC522 mfrc522(SS_PIN, RST_PIN);

byte knownUID[] = {0xE3, 0x33, 0xEB, 0x2C};

bool remoteUnlock = false;
```

```
void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();

  pinMode(led, OUTPUT);
  pinMode(buz, OUTPUT);
  pinMode(LOCK_PIN, OUTPUT);
  digitalWrite(LOCK_PIN, HIGH);  // Locked by default

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
 configTime(19800, 0, "pool.ntp.org");  // GMT+5:30 (IST). Change offset if needed


  Serial.println("Ready. Scan your RFID card...");
}

void loop() {
  Blynk.run();

  // Remote unlock via Blynk switch
  if (remoteUnlock) {
    digitalWrite(LOCK_PIN, LOW);   // Unlock
    digitalWrite(led, HIGH);
    return;  // Skip RFID check if remotely unlocked
```

```
} else {
  digitalWrite(LOCK_PIN, HIGH);  // Locked
  digitalWrite(led, LOW);
}

digitalWrite(buz, LOW);  // Default: buzzer off

// Wait for a card
if (!mfrc522.PICC_IsNewCardPresent()) return;
if (!mfrc522.PICC_ReadCardSerial()) return;

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

// Check UID
if (compareUID(mfrc522.uid.uidByte, knownUID, mfrc522.uid.size)) {
  Serial.println("✅ Access Granted - Unlocking for 5 seconds");
  Blynk.virtualWrite(V0, "✅ Abinaya! unlocked the door at " + getCurrentTime());
  digitalWrite(LOCK_PIN, LOW);  // Unlock
  digitalWrite(led, HIGH);
```

```
    delay(5000);                // Stay unlocked
  } else {
    Serial.println("❌ Access Denied - Staying locked");
    Blynk.virtualWrite(V0, "❌ Unauthorized card scanned at " + getCurrentTime());

    digitalWrite(buz, HIGH);
    delay(2000);
  }

  // Stop communication
  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();

  delay(500);  // Small gap before next scan
}
String getCurrentTime() {
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    return "⏱ time error";
  }
  char timeStr[20];
  strftime(timeStr, sizeof(timeStr), "%H:%M:%S", &timeinfo);
  return String(timeStr);
}
```

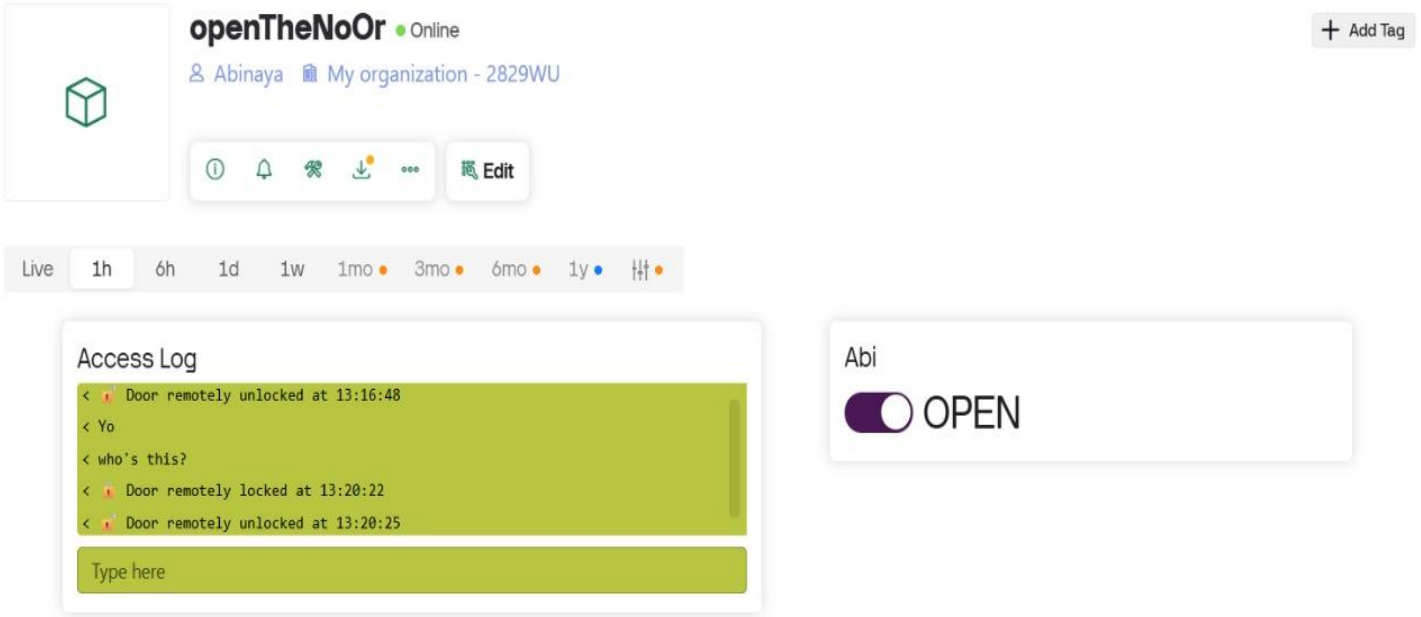
```
// UID checker
bool compareUID(byte* scannedUID, byte* validUID, byte len) {
  for (byte i = 0; i < len; i++) {
    if (scannedUID[i] != validUID[i]) return false;
  }
  return true;
}


// Blynk Switch Handler (V1)
BLYNK_WRITE(V1) {
  int value = param.asInt();
  remoteUnlock = (value == 1);


if (remoteUnlock) {
  Serial.println("🔓 Door remotely unlocked (switch ON)");
  Blynk.virtualWrite(V0, "🔓 Door remotely unlocked at " + getCurrentTime());
} else {
  Serial.println("🔒 Door remotely locked (switch OFF)");
  Blynk.virtualWrite(V0, "🔒 Door remotely locked at " + getCurrentTime());
}
}
```

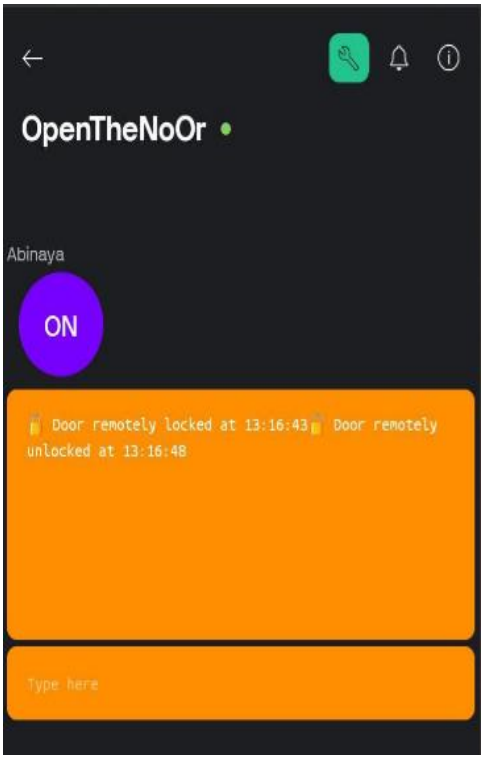
# OUTPUT:

## LOCKING AND UNLOCKING OF THE DOOR USING CLOUD IN WEBSITE




## ACCESSING THE DOOR LOCK USING USER1'S PHONE WITH THE HELP OF BLYNK APP

**ACCESSING THE DOOR LOCK USING USER2'S PHONE WITH THE HELP OF BLYNK APP**




## HOW CAN THIS PROJECT BE IMPROVED

**1. Smart Schedule Check:**

-System cross-verifies door unlock time against stored class timetables

-Priority Alert Trigger - During class hours: activates loud buzzer + repeated Blynk alerts to all roommates

**2. Security Enhancements:**

- Add password protection in Blynk.
- Encrypt Wi-Fi and communication channels.

**3. Camera Integration:**

- Capture image of unauthorized users.
- Send it to email or app

**4. Battery Backup and Power Monitoring:**

- Add a battery for continuous operation during power cuts.
- Monitor voltage levels for hardware protection.

**5. Multi-Factor Authentication (MFA) Systems**

(A) Fingerprint Sensor (e.g., R307)

(B) Face Recognition (ESP32-CAM)

(C) Keypad + RFID Hybrid

## CONCLUSION

The Smart Door Lock System using RFID and Blynk successfully demonstrates the fusion of embedded systems with IoT technology for secure, flexible, and convenient access control. It automates door locking mechanisms, provides real-time monitoring, and introduces smart control through mobile integration. With a few enhancements, this project can evolve into a complete, scalable smart access system suitable for real-world deployment.