

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

const int SS_PIN = 10;
const int RST_PIN = 9;

MFRC522 rfid(SS_PIN, RST_PIN);

const byte ROWS = 4;
const byte COLS = 4;

char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {A5, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

const int MAX_CARDS = 5;

byte cardUIDs[MAX_CARDS][4];
String cardPins[MAX_CARDS];
int cardCount = 0;

int buzzer = A4;
int gled = A3;
int rled = A2;
```

```

void setup()
{
    pinMode(buzzer, OUTPUT);
    pinMode(gled, OUTPUT);
    pinMode(rled, OUTPUT);
    digitalWrite(buzzer, LOW);

    Serial.begin(9600);
    SPI.begin();
    rfid.PCD_Init();

    showMenu();
}

void loop()
{
    char key = keypad.getKey();
    if (!key)
    {
        return;
    }

    digitalWrite(buzzer, HIGH);
    delay(20);
    digitalWrite(buzzer, LOW);

    while (keypad.getKey())
    {

    }

    if (key == '1')
    {
        addCard();
    }
    else if (key == '2')
    {
        accessCard();
    }
    else if (key == '4')

```

```

    {
        removeCard();
    }
    showMenu();
}

void showMenu()
{
    Serial.println("\n--- MENU ---");
    Serial.println("1: Add new card");
    Serial.println("2: Access card"); //menu
that prints
    Serial.println("4: Remove card");
    Serial.println("Press a key...");
}

void addCard()
{
    if (cardCount >= MAX_CARDS)
    {
        Serial.println("Storage full");
        return;
    }

    Serial.println("Scan card to add...");

    byte newUID[4];
    waitForCard(newUID); // The
card has 4 Bytes

    for (int i = 0; i < cardCount; i++)
    {
        bool duplicate = true;
        for (int j = 0; j < 4; j++)
        {
            if (newUID[j] != cardUIDs[i][j])
            {
                duplicate = false;
            }
        }
    }
}

```

```

        if (duplicate)
        {
            Serial.println("Card already added!");
            return;
        }
    }

    memcpy(cardUIDs[cardCount], newUID, 4);

    Serial.print("Card scanned successfully. UID: ");
    printUID(newUID);

    Serial.println("Create 4-digit PIN:");
    cardPins[cardCount] = readPIN();

    cardCount++;

    Serial.println("Card added successfully");
    digitalWrite(gled, HIGH);
    delay(1000);
    digitalWrite(gled, LOW);
}

void accessCard()
{
    Serial.println(" ");
    Serial.println("Please Scan The Card You Want To Access...");

    int eh = waitForExistingCard();

    if (eh == -1)
    {
        return;
    }

    Serial.println("Enter PIN:");

```

```

    if (readPIN() == cardPins[eh])
    {
        Serial.println("Access granted");
        digitalWrite(gled, HIGH);
        delay(1000);
        digitalWrite(gled, LOW);
    }
    else
    {
        Serial.println("Access denied");
        digitalWrite(buzzer, HIGH);
        digitalWrite(rled, HIGH);
        delay(1600);
        digitalWrite(buzzer, LOW);
        digitalWrite(rled, LOW);
    }
}

void removeCard()
{
    Serial.println("Scan card to remove...");

    int index = waitForExistingCard();

    if (index == -1)
    {
        return;
    }

    Serial.println("Enter PIN:");

    if (readPIN() != cardPins[index])
    {
        Serial.println("Wrong PIN");
        digitalWrite(buzzer, HIGH);
        digitalWrite(rled, HIGH);
        delay(1600);
        digitalWrite(buzzer, LOW);
        digitalWrite(rled, LOW);
        return;
    }
}

```

```

    for (int i = index; i < cardCount - 1; i++)
    {
        memcpy(cardUIDs[i], cardUIDs[i + 1], 4);
        cardPins[i] = cardPins[i + 1];
    }

    cardCount--;

    Serial.println("Card removed successfully");
}

void waitForCard(byte dest[4])
{
    while (true)
    {
        if (rfid.PICC_IsNewCardPresent())
        {
            if (rfid.PICC_ReadCardSerial())
            {
                for (int i = 0; i < 4; i++)
                {
                    dest[i] = rfid.uid.uidByte[i];
                }

                rfid.PICC_HaltA();
                rfid.PCD_StopCrypto1();
                delay(200);
                break;
            }
        }

        delay(50);
    }
}

int waitForExistingCard()
{
    byte uid[4];

```

```

while (true)
{
    if (rfid.PICC_IsNewCardPresent())
    {
        if (rfid.PICC_ReadCardSerial())
        {
            for (int i = 0; i < 4; i++)
            {
                uid[i] = rfid.uid.uidByte[i];           //stores the card
in the array
            }

            for (int i = 0; i < cardCount; i++)
            {
                bool match = true;

                for (int j = 0; j < 4; j++)
                {
                    if (uid[j] != cardUIDs[i][j])
                    {
                        match = false;
                    }
                }

                if (match)
                {
                    rfid.PICC_HaltA();
                    rfid.PCD_StopCrypto1();
                    delay(200);

//
Had to look this up to match the UID also the in built functions
                    Serial.print("Card scanned successfully. UID: ");
                    printUID(uid);
                    return i;
                }
            }

            rfid.PICC_HaltA();
            rfid.PCD_StopCrypto1();
            delay(200);

```

```

        Serial.println("Card not found, try again.");
        digitalWrite(buzzer, HIGH);
        digitalWrite(rled, HIGH);
        delay(1600);
        digitalWrite(buzzer, LOW);
        digitalWrite(rled, LOW);
    }
}

    delay(50);
}
}

```

```

void printUID(byte uid[4])
{
    for (int i = 0; i < 4; i++)
    {
        Serial.print(uid[i], HEX);
        Serial.print(" ");
    }
}

```

```

    Serial.println();
}

```

```

String readPIN()
{
    String pin = "";

    while (pin.length() < 4)
    {
        char key = keypad.getKey();

        if (key >= '0' && key <= '9')
        {

            digitalWrite(buzzer, HIGH);
            delay(40);
            digitalWrite(buzzer, LOW);

            pin += key;
        }
    }
}

```

```
        Serial.print("*");  
    }  
}  
  
Serial.println();  
return pin;  
}
```