🔧 Detailed Project Explanation: RFID Access Control System

1. Device Specifications and Functionalities

□ Arduino UNO (Microcontroller)

- **Specifications:**
  - Microcontroller: ATmega328P
  - Operating Voltage: 5V
  - Digital I/O Pins: 14 (6 PWM outputs)
  - Analog Inputs: 6
  - Clock Speed: 16 MHz
  - Communication Interfaces: UART, SPI, I^2C
- **Functionality:**
  - Acts as the **central processing unit** of the system.
  - Receives data from the RFID module via the SPI interface.
  - Compares the scanned card's UID with predefined authorized IDs.
  - Controls the **LED indicators**, **buzzer**, and **lock output** based on access validation.
  - Manages input from the **manual push button** for emergency/universal access.
- **What it does not do:**
  - It doesn't store large databases of RFID tags (limited EEPROM memory).
  - It doesn't handle complex networking (no Wi-Fi or GSM by default).

📶 MFRC522 RFID Reader Module

- **Specifications:**
  - Operating Voltage: 3.3V
  - Frequency: 13.56 MHz
  - Communication Interface: SPI
  - Read Range: 2–5 cm
  - Compatible RFID Tags: MIFARE 1K, 4K, and other ISO/IEC 14443-A cards
- **Functionality:**
  - Detects nearby RFID cards/tags and reads their **unique identifier (UID)**.
  - Sends UID data to Arduino via SPI communication for verification.
  - Operates at a short range to prevent unauthorized distance-based access.
- **What it does not do:**
  - Cannot write or store access records by itself.
  - Doesn't handle authentication logic — it only reads tag IDs.

💡 LED Indicators (Green & Red LEDs)

- **Specifications:**
  - Type: Standard 5mm LEDs
  - Voltage: 2V (typically with 220Ω current-limiting resistor)
- **Functionality:**
  - Green LED: Lights up when access is granted.
  - Red LED: Lights up when access is denied.
- **What they do not do:**
  - They don’t provide audio feedback or store any status information.

🔊 Buzzer

- **Specifications:**
  - Operating Voltage: 3V–5V
  - Type: Piezoelectric buzzer
- **Functionality:**
  - Emits short beeps to indicate events:
    - **Two short beeps:** Access granted
    - **Long continuous beep:** Access denied
  - Provides **audible feedback** for better user interaction.
- **What it does not do:**
  - It doesn’t differentiate users or provide any data — purely a feedback component.

□ Push Button

- **Specifications:**
  - Simple tactile switch with pull-down configuration
  - Connected to digital pin for manual input
- **Functionality:**
  - Simulates a manual override (for admin or emergency use).
  - When pressed, grants temporary access without scanning an RFID card.
- **What it does not do:**
  - It cannot store or verify identity — only triggers a bypass condition.

🔒 Door Lock (Relay or Output Pin)

- **Specifications:**
  - In prototype: Simulated using an LED or DC motor output.
  - Real-world: Controlled via relay or solenoid lock (12V).
- **Functionality:**
  - Activates when valid RFID tag is detected or button is pressed.

    - Unlocks door for a fixed duration, then returns to locked state.
- **What it does not do:**
    - It doesn’t perform timing control by itself — Arduino handles timing logic.

. Device Interlinking and System Advantages

- The **Arduino UNO** acts as the **core controller**, interfacing with all other devices.
- The **RFID Reader** communicates through the **SPI bus**, ensuring fast and reliable UID transfer.
- The **LEDs and buzzer** provide immediate feedback controlled by Arduino’s digital pins.
- The **push button** and **lock** connect directly to I/O pins for simple, synchronous operation.

Advantages of this setup:

- Modular design — easy to expand with GSM, Wi-Fi, or display modules.
- Real-time response due to SPI communication.
- Low power consumption and affordable components.
- Simple wiring and reliable performance for prototype and real-world use.

. Limitations and Considerations

- **Limited UID storage** — only a few RFID tags can be hardcoded unless external memory is added.
- **Short read range** (2–5 cm) — intentional for security, but not suitable for high-speed scanning.
- **No remote monitoring** — system works offline unless an IoT module is integrated.
- **Lock simulation** — real deployment needs a 12V relay and external power supply for the lock.

. Suitability for the Project Requirements

- The project aims for a **low-cost, secure, and user-friendly** access control system.
- All selected components are:
    - **Affordable and easily available**
    - **Compatible with Arduino IDE and libraries**
    - **Low-power and compact**, ideal for embedded use
    - **Highly reliable** for repeated scanning and switching operations

Device	Reason for Selection
Arduino UNO	Easy to program, affordable, wide community support, sufficient GPIO pins for all peripherals.
MFRC522 RFID Reader	Reliable, compact, SPI-compatible, and low-cost compared to other RFID modules (like PN532).
LEDs & Buzzer	Simple yet effective visual and audio indicators.
Push Button	Provides manual testing and emergency access functionality.
Relay/Lock Output	Directly interfaces with physical locking mechanisms for real-world control.