SMART DOOR LOCK SYSTEM

By

Harveerdan Gadhavi (23MCA034)

Kishan Gajera (23MCA036)

Ujjval Gandhi (23MCA039)

Under Guidance

of

Internal Guide

Dr. Amisha Shingala

Submitted to

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications

CHARUSAT Changa

Nov-Dec 2024

Accredited with Grade A+ by NAAC,

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Changa

Acknowledgement

Knowledge in itself is a continuous process. At this moment of our substantial enhancement, We rarely find words to express our gratitude towards those who were constantly involved with us.

The completion of any inter disciplinary project depends upon coordination, cooperation and combined efforts of several resources of knowledge, creativity, skill, energy and time. The work being accomplished now, we feel our sincerest urge to recall and knowledge through these lines, trying our best to give full credit wherever it deserves.

We would like to thank our project guide **Dr. Amisha Shingala,** I/C Principal **Dr. Dharmendra Patel** and I/C Dean **Dr. Sanskruti Patel** who advised and gave us moral support through the duration of our project. Without their constant encouragement we could not have been able to achieve what we have.

It's our good fortune that we had support and well wishes of many. We are thankful to all and those names which have been forgotten to acknowledge here but contributions have not gone unnoticed.

With Sincere Regards,

Harveerdan Gadhavi (23MCA034)

Kishan Gajera (23MCA036)

Ujjval Gandhi (23MCA039)

Abstract

This smart door lock system combines hardware components, including an Arduino, RFID reader, fingerprint sensor, LCD, and buzzer, with a Flutter-based mobile application for secure, contactless access control. The system uses MQTT (Message Queuing Telemetry Transport) protocol to facilitate reliable, real-time communication between the Flutter app and the hardware over Wi-Fi. Through the app, users can register fingerprints, manage RFID access cards, and monitor entry logs remotely. MQTT enables efficient data exchange, ensuring low-latency updates and secure control of the door lock from any location. This project enhances security for residential and commercial applications, supporting multiple users and customizable access permissions with scalability for larger deployments.

Index

NO	Subject	Page No.
1	Project Profile	5
2	Introduction to tools	7
3	System Design 3.1 Circuit Diagram 3.2 Flow Chart 3.3 Use Case Diagram 3.4 Activity Diagram	23
4	Screen Layout	28
5	System Study • Existing Study • Proposed System • Scope of the Proposed System • Aim and objective Proposed System • What we Actually did in the project?	31
6	Feasibility Study • Operational Feasibility • Technical Feasibility • Economical Feasibility	34
7	System Testing	36
8	Future Enhancement	37
9	References	38

PROJECT PROFILE

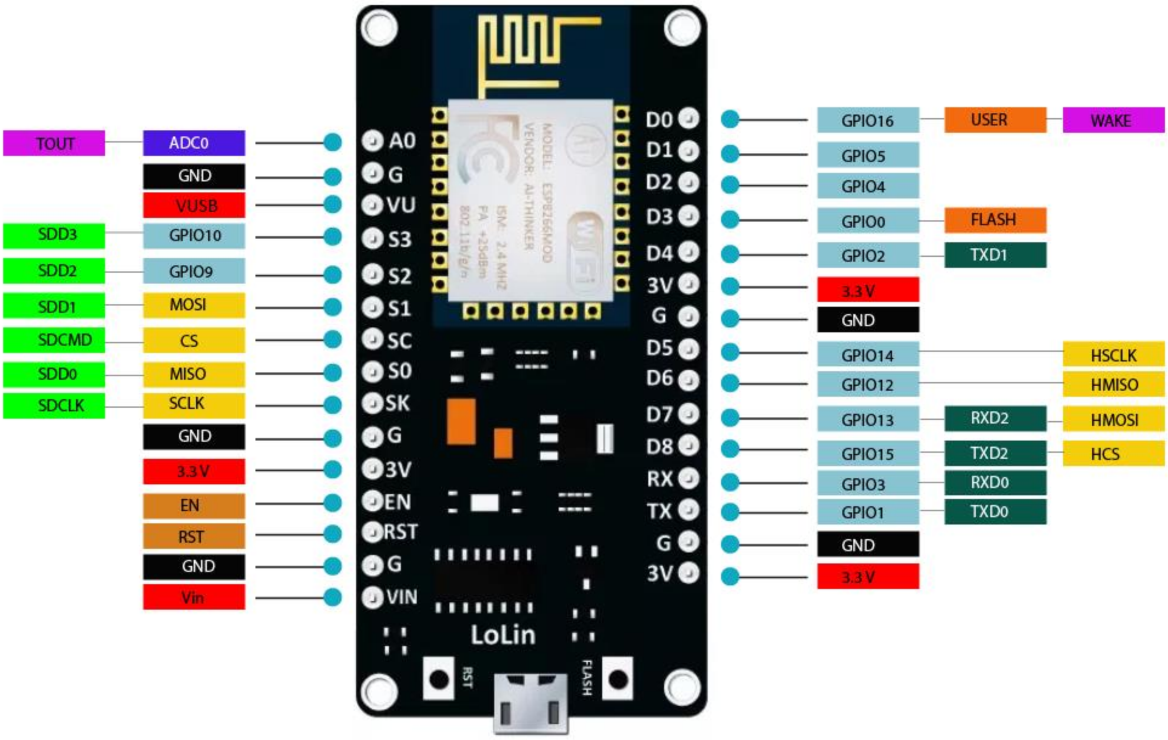
Project name	Smart door lock system
Type of Application	Hardware and software based Project
Project Description	This project focuses on developing a smart door lock system that enhances home and office security by enabling users to control door access through a mobile application built using Flutter.
Front end	Flutter, Embedded C
Back end	Dart, MQTT
Database	Firebase
Team size	3
Tools used	Arduino IDE, Android Studio, Firebase

INTRODUCTION TO TOOLS

Introduction to Tools

➢ Front End Tool :

- **ESP32 :**

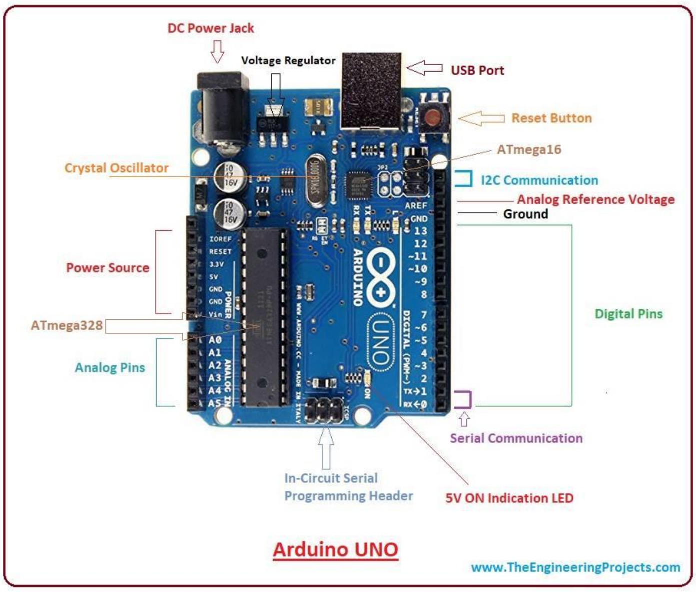


The ESP32 is a versatile microcontroller, widely used for IoT applications due to its integrated Wi-Fi and Bluetooth capabilities. Developed by Espressif Systems, it features a dual-core processor with speeds up to 240 MHz, providing high performance for a variety of tasks. Its energy-efficient design supports multiple low-power modes, making it ideal for battery-powered devices.

The ESP32 supports a range of communication protocols, including SPI, I2C, UART, and PWM, which allows it to interface with a wide array of sensors and actuators. It also has up to 520 KB of SRAM and support for external flash storage, enabling more complex programs and data handling.

Its built-in Bluetooth 4.2 and BLE support, along with Wi-Fi, make it perfect for connected devices that need to communicate wirelessly, such as smart locks, home automation systems, and wearables. In your smart door lock system, the ESP32 can control the locking mechanism, process user commands, and provide secure communication with your Flutter app, ensuring real-time control and monitoring. Its cost-effectiveness and flexibility make it a popular choice for both hobbyist projects and professional IoT solutions.

✓ Arduino Uno :



The Arduino Uno is a versatile and widely used microcontroller board, ideal for beginners and experts in electronics and prototyping. It is based on the ATmega328P microcontroller and features 14 digital I/O pins (6 supporting PWM), 6 analog input pins, a 16 MHz quartz crystal, a USB port, a power jack, and a reset button.

The board can be powered via USB or an external power source (7-12V) and is programmed using the Arduino IDE, which provides a simple C++-based environment with extensive libraries for hardware interfacing. With support for communication protocols like I2C, SPI, and UART, it easily connects to sensors, displays, motors, and other components.

The Uno has 32 KB flash memory (0.5 KB used by the bootloader), 2 KB SRAM, and 1 KB EEPROM. It operates at 5V logic levels and is compatible with numerous shields for extended functionality.

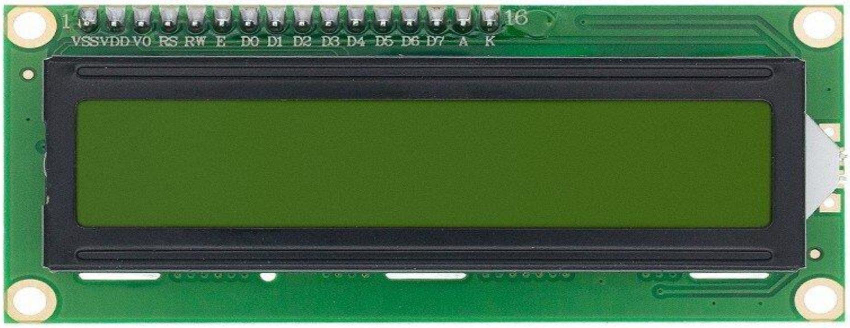
Commonly used in IoT, robotics, and automation projects, the Arduino Uno’s open-source design and large community make it a favorite for creating innovative solutions, from basic LED control to advanced systems. Its reliability and ease of use make it a cornerstone of modern electronics prototyping.

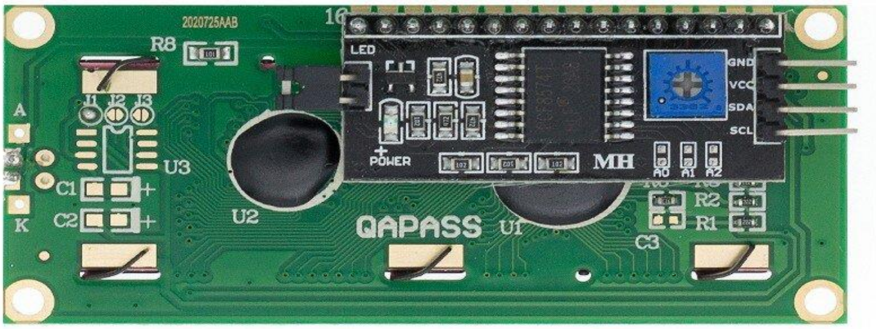
✓ **Fingerprint Scanner (R307)** :

The R307 fingerprint scanner is an optical biometric sensor used for secure access control. It captures and stores fingerprint data, comparing scanned fingerprints with saved templates for identification. With a storage capacity of up to 1000 fingerprints, it uses UART communication to interface with microcontrollers like the ESP32.

To use it in a project, the R307 connects via serial pins: TX, RX, VCC, and GND. When a fingerprint is scanned, the sensor processes it, converts it into a template, and stores it. For matching, the sensor compares new scans with stored templates to verify identity. It supports 1:1 verification (comparing one fingerprint) and 1 matching (searching through a database).This makes the R307 ideal for projects like smart locks, ensuring secure and convenient access control.

• I2C LCD Display :

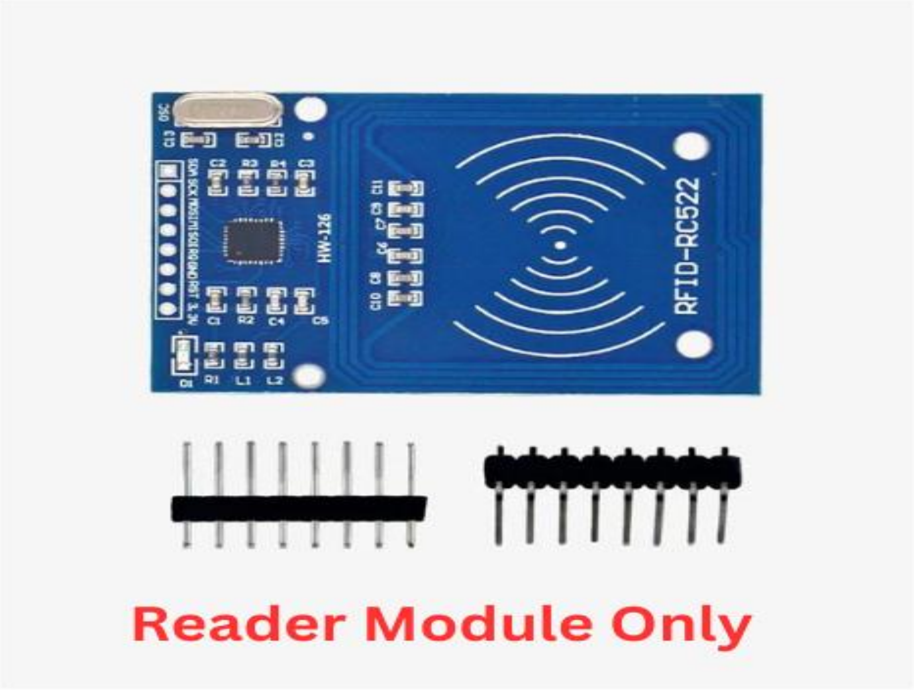




An I2C LCD display is a commonly used module that simplifies the connection to microcontrollers by using only two communication lines: SDA (data) and SCL (clock). It typically comes in 16x2 or 20x4 formats, displaying characters and text, ideal for embedded systems and IoT projects like a smart door lock.Using the I2C interface significantly reduces wiring complexity compared to a standard parallel LCD. It connects to the microcontroller using just four wires: VCC, GND, SDA, and SCL. The display can show real-time information like door status, access logs, or error messages.

In the Arduino IDE, you can use the **LiquidCrystal_I2C** library for easy interaction. Once initialized, the LCD can be controlled with simple commands like `lcd.print()` to display text and `lcd.setCursor()` to position the text on different rows and columns. Additionally, the backlight can be controlled to save power when the display is not in use.

- **RFID (RC522) Redear module** :



The RFID RC522 reader module is a low-cost, 13.56 MHz RFID reader commonly used for reading and writing RFID tags. It communicates with microcontrollers like ESP32 or Arduino through the SPI protocol. This module is ideal for projects like access control, smart locks, and attendance systems.

The RC522 reads RFID tags that follow the ISO 14443A standard, such as MIFARE cards. With a typical reading range of a few centimeters, it's perfect for short-range secure identification. It requires minimal wiring, with connections for VCC, GND, SDA, SCK, MOSI, MISO, and RST.

In a project, the module can be used to authenticate users by scanning RFID tags. For example, when an authorized tag is detected, the system can trigger actions like unlocking a door. Using the **MFRC522** library in Arduino IDE, the RFID reader is easy to set up, and code can be written to read the tag’s UID, which can be used for security and identification purposes.

➢ LM2596 DC – DC Step Down :

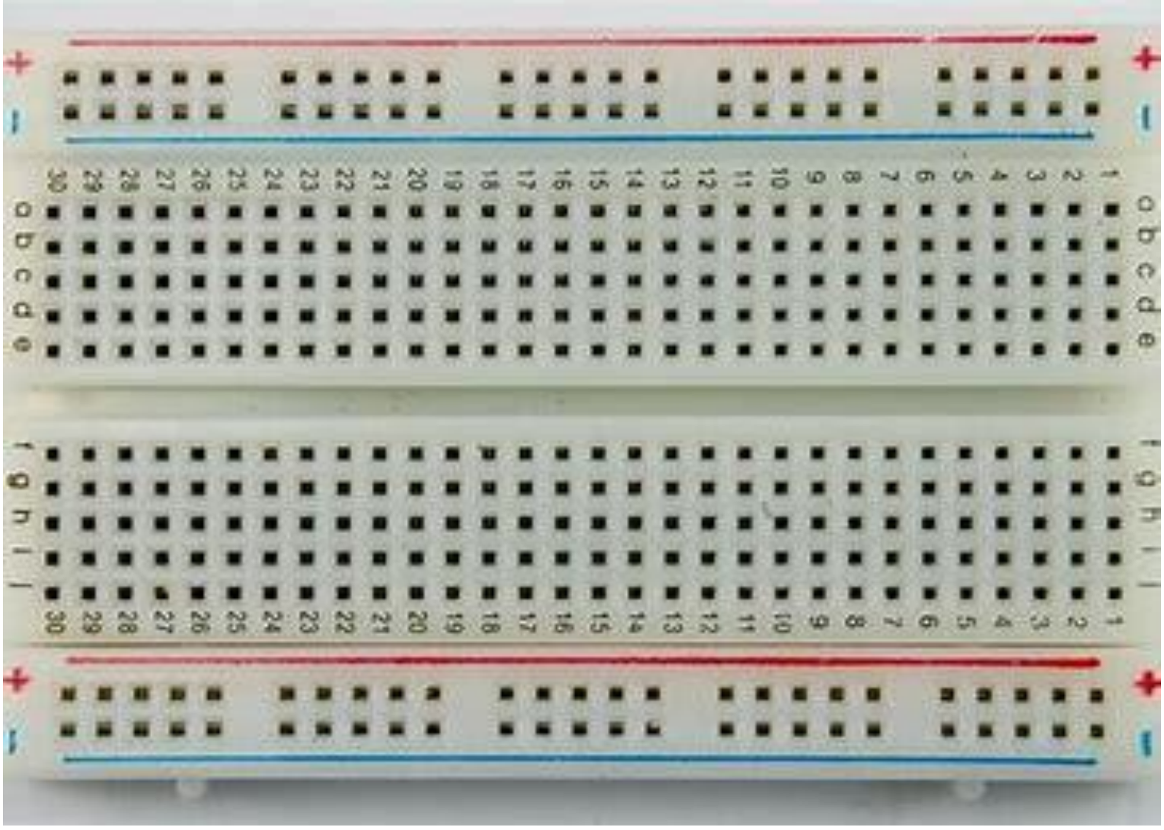
A relay module is an essential component in electronic and IoT projects, allowing low-power microcontrollers like Arduino, Raspberry Pi, or ESP32 to control higher power devices safely. The relay itself acts as an electrically operated switch, isolating the high-power circuit from the control circuit. When the control circuit sends a signal (usually from a GPIO pin on the microcontroller), the relay activates and either opens or closes the circuit connected to its terminals.

Relay modules typically come with one or multiple channels (1, 2, 4, 8, etc.), allowing control over multiple devices. Each channel has an onboard relay that can switch devices operating at higher voltages, such as AC appliances (lights, fans) or DC motors. The module includes transistors, diodes, and optocouplers to protect the microcontroller from voltage spikes and ensure reliable operation.

Relay modules are used in applications such as home automation, where you can control appliances remotely, or in industrial systems, where automation requires switching heavy machinery. Most relay modules are rated for 5V or 12V control, with a trigger input that activates the relay when connected to a high or low signal, depending on the relay type.

Safety is crucial when working with relay modules, especially with high-voltage AC loads. Proper insulation, grounding, and careful handling of wiring are essential. The module should be rated for the specific voltage and current of the devices being controlled.

➢ Breadboard :



A breadboard is a reusable, modular platform for prototyping electronic circuits without soldering. It consists of a grid of holes connected by conductive strips, allowing components like resistors, capacitors, and integrated circuits to be inserted easily. Breadboards typically have a power rail for connecting power supplies, making it convenient to distribute voltage to various parts of a circuit. They are essential tools for experimenting and testing circuit designs in a straightforward manner. Ideal for beginners and professionals alike, breadboards facilitate rapid development and iteration of electronic projects.

➢ Battery :

A 12V battery is a popular power source used in automotive, renewable energy, and portable electronic applications. Typically consisting of six cells, it provides a nominal voltage of 12V and is available in various chemistries, including lead-acid and lithium-ion. These batteries can come in different capacities, measured in amp-hours (Ah), to suit various energy needs. Many are rechargeable, making them cost-effective and environmentally friendly. Overall, 12V batteries are versatile and reliable, serving as essential components in numerous devices and systems.

traditional switches. Overall, these modules are versatile and effective in a wide range of electronic projects.

➢ Buzzer :

A buzzer is an electronic device that produces sound when an electric current passes through it, commonly used in alarms, notifications, and indicators. They come in two main types: active and passive, with active buzzers generating sound at a specific frequency and passive buzzers producing sound based on the input frequency. Buzzers are easy to integrate into circuits, requiring only a power supply and a signal input, making them popular in DIY projects and electronics. They are widely used in applications like timers, alarms, and alerts in home automation systems. Overall, buzzers are simple yet effective components for generating audio signals in various electronic applications.

➢ Jumper Cable :

Jumper cables are insulated wires with alligator clips at each end, used to connect two electrical devices, typically for providing power or transferring signals. They are commonly used to connect various components in electronic projects. Available in various lengths and gauge sizes, jumper cables are designed for high current and reliability. Their ease of use makes them essential tools for both automotive and electronics enthusiasts. Overall, jumper cables are versatile and practical for quickly establishing temporary connections in various applications.

➢ Back End Tools :

• Embedded C :

Embedded C is an extension of the C programming language tailored for programming microcontrollers and embedded systems. It allows developers to write efficient code that directly interacts with hardware components. One of its key features is the ability to access hardware registers, enabling precise control over peripherals like timers, ADCs, and GPIOs.

In Embedded C, efficient memory management is crucial due to limited resources in embedded devices. The language supports various data types, including bit fields and volatile types, which are essential for real-time applications. It also provides mechanisms for interrupt handling, allowing systems to respond to external events promptly.

Additionally, Embedded C is designed for portability, enabling code to be adapted for different hardware platforms with minimal changes. The use of preprocessor directives, such as `#define`, helps manage configurations and constants effectively.

A simple example of Embedded C is blinking an LED connected to a microcontroller pin. The program initializes the pin as an output and toggles it on and off in a loop, showcasing basic GPIO manipulation. Overall, Embedded C is a powerful tool for developing applications in various domains, including automotive, industrial automation, and consumer electronics.

➢ Flutter :

Flutter is an open-source UI software development kit created by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It uses the Dart programming language, enabling developers to create high-performance, visually appealing apps with a rich set of pre-designed widgets.

One of Flutter's key features is its hot reload capability, allowing developers to see code changes in real time without losing the app's state. This significantly speeds up the development process and enhances productivity. Flutter's widget-based architecture means that everything is a widget, making it easy to compose complex UIs from simpler components.

Flutter provides a rich set of Material Design and Cupertino (iOS-style) widgets, ensuring that apps can have a native look and feel on both Android and iOS platforms. It also includes a powerful animation library, allowing for smooth transitions and engaging user experiences.

For backend integration, Flutter supports various options, including REST APIs and Firebase. Additionally, Flutter has a growing ecosystem of packages and plugins available via the Dart Package Repository, making it easier to add functionality like maps, payments, and device sensors.

The community surrounding Flutter is vibrant and rapidly growing, offering numerous resources, tutorials, and libraries to help developers. Overall, Flutter is a versatile framework that simplifies the app development process while providing the tools to create high-quality applications.

➢ Firebase :

Firebase is a powerful backend-as-a-service platform provided by Google that offers tools like Google Authentication and a real-time database to build and manage modern applications efficiently.

Google Authentication allows seamless integration of Google's secure login system into apps, enabling users to authenticate with their Google accounts. This authentication process supports multiple platforms, including Android, iOS, and web. It is easy to implement using the Firebase Authentication SDK, which handles complex tasks like token generation, account linking, and session management. Developers can also combine Google Authentication with other methods, such as email/password or phone number, for multi-method authentication.

Firebase Database provides two options: the Realtime Database and Firestore, both offering cloud-hosted, NoSQL databases that store and sync data in real-time. The Realtime Database enables real-time synchronization between connected clients, making it ideal for collaborative applications like chat apps or live dashboards. Firestore, on the other hand, supports richer querying, offline capabilities, and scalable structures.

Both the database options integrate seamlessly with Firebase Authentication, enabling secure, role-based access to data. Security rules allow developers to define granular permissions based on user identity and data context, ensuring only authorized users can access specific resources.

With Firebase, developers can create highly interactive, user-centric applications by combining Google Authentication for secure user login and Firebase Database for dynamic, real-time data handling. Its ease of use, scalability, and cross-platform support make Firebase an essential tool for modern app development.

➢ MQTT :

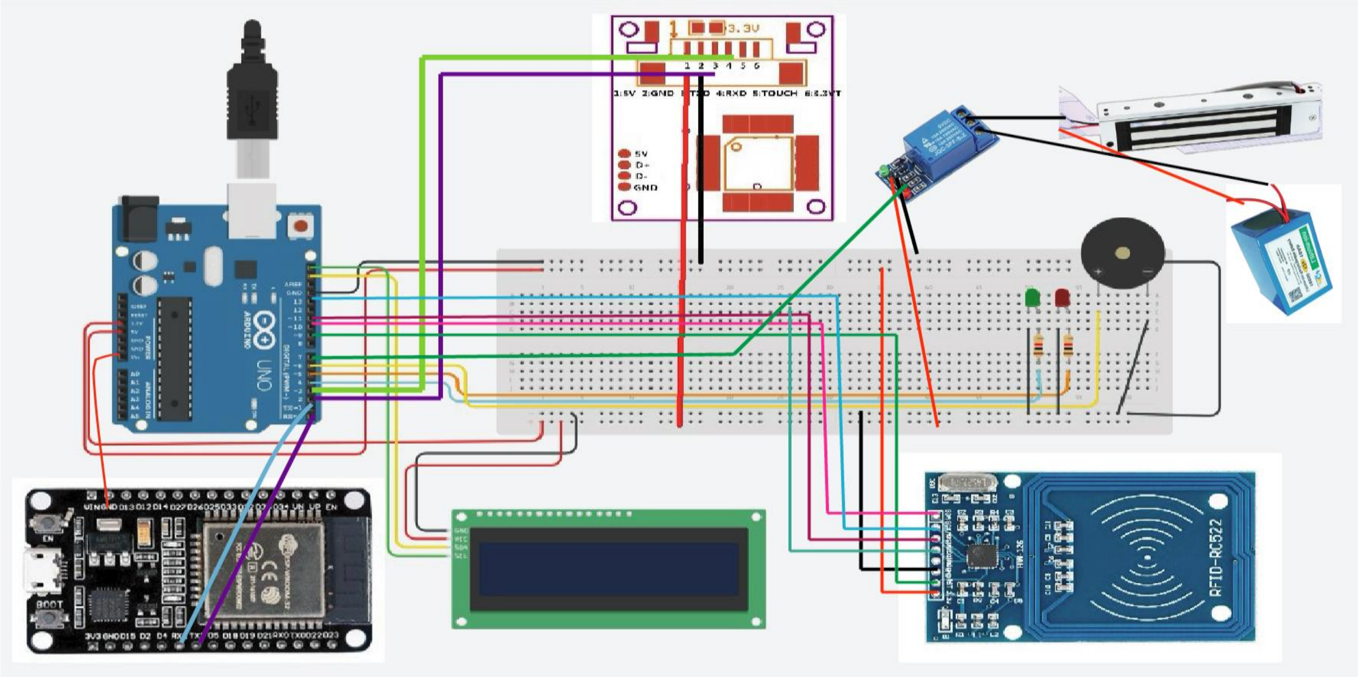
MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for resource-constrained devices and unreliable networks, making it a cornerstone of IoT communication. It operates on a publish-subscribe model, where devices communicate through a central broker. A publisher sends messages to specific topics, while a subscriber listens to topics of interest, enabling decoupled and efficient communication. For instance, a temperature sensor may publish data to the topic home/livingroom/temperature, and any device subscribed to this topic receives real-time updates.

MQTT supports three Quality of Service (QoS) levels: QoS 0 (at most once), QoS 1 (at least once), and QoS 2 (exactly once), ensuring reliable message delivery tailored to application needs. Features like retained messages allow the broker to store the latest message on a topic for new subscribers, while last will messages notify connected devices of unexpected client disconnections. With its lightweight design, MQTT minimizes bandwidth usage, making it ideal for microcontrollers and low-power devices.

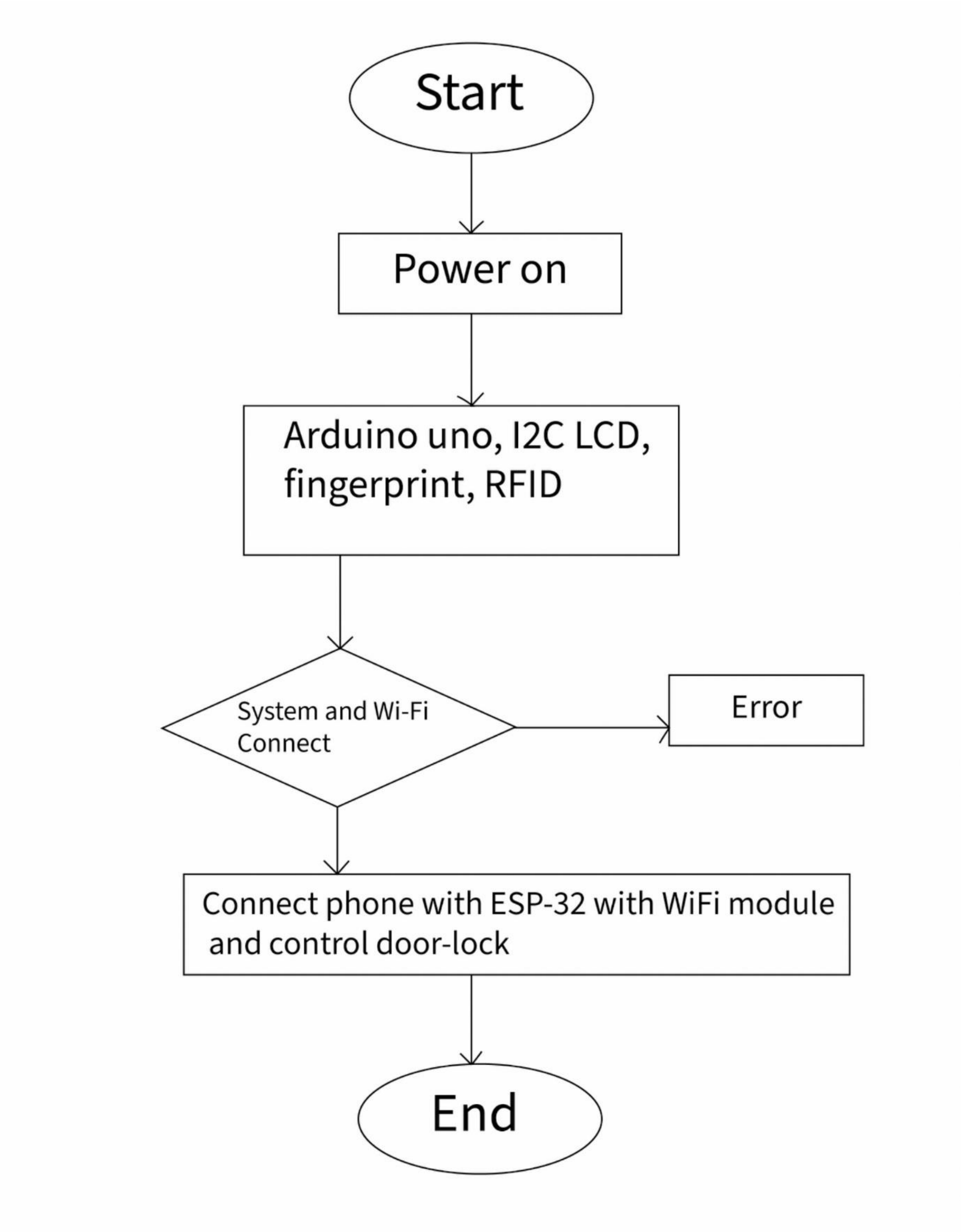
Applications range from smart homes and industrial automation to healthcare and real-time data monitoring. Secure communication is achieved through TLS/SSL encryption and authentication mechanisms, ensuring reliable and safe data exchange between publishers, subscribers, and the broker.

SYSTEM DESIGN

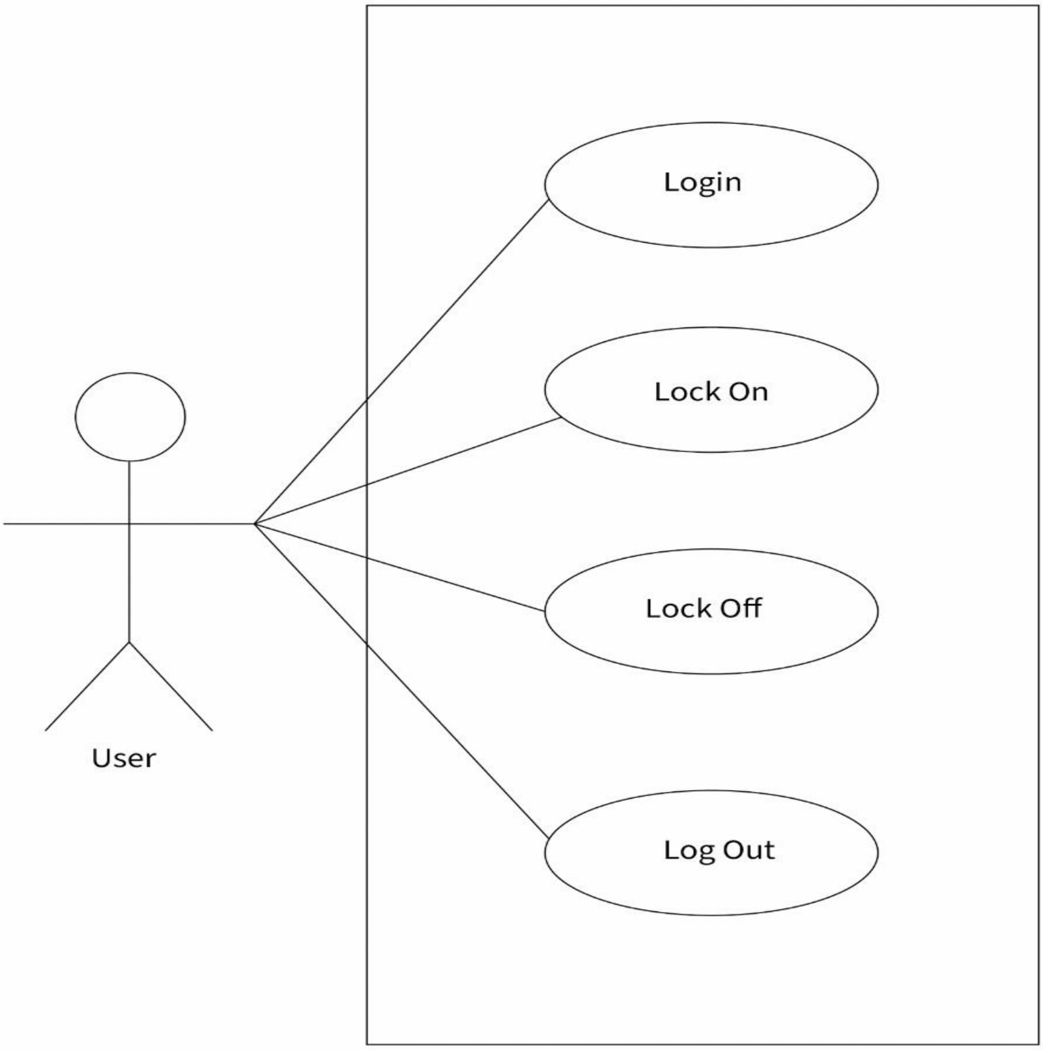
Circuit Diagram :



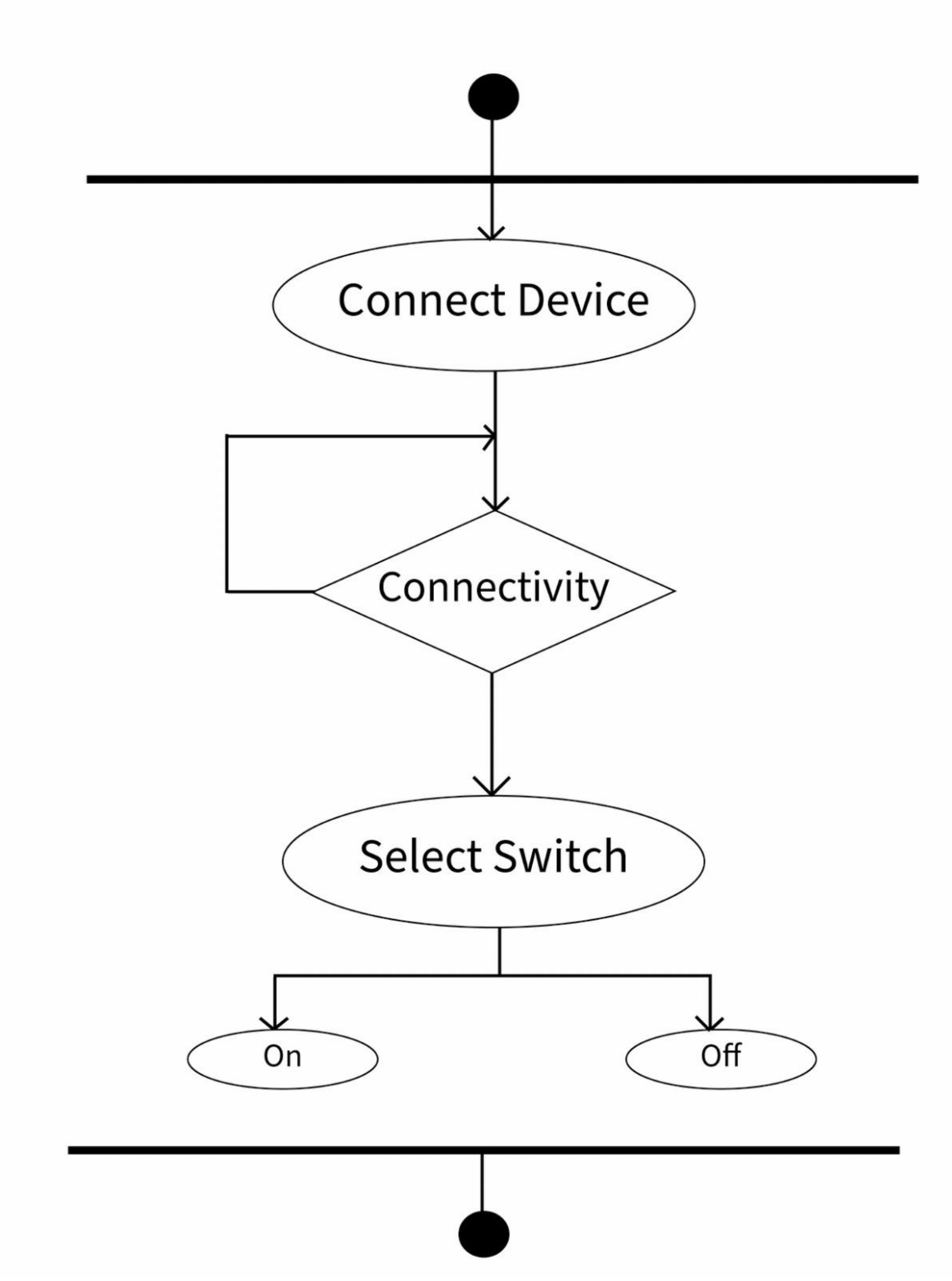
Flow Chart :


Start
Power on
Arduino uno, I2C LCD, fingerprint, RFID
System and Wi-Fi Connect
Error
Connect phone with ESP-32 with WiFi module and control door-lock
End

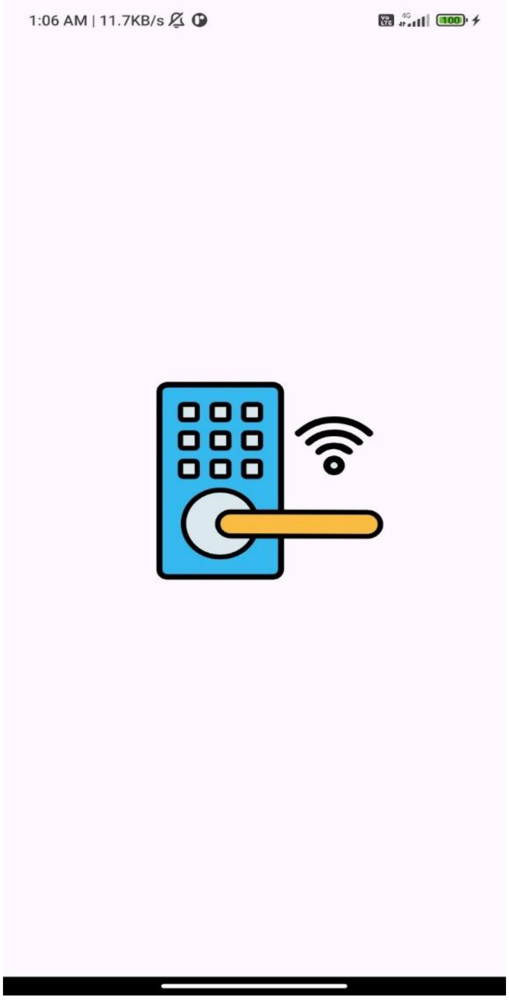
Use case Diagram :

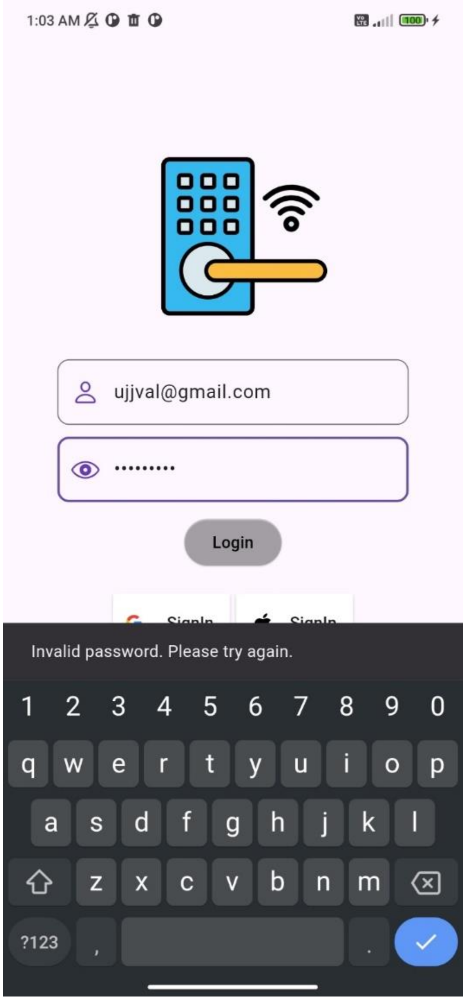


Activity Diagram :

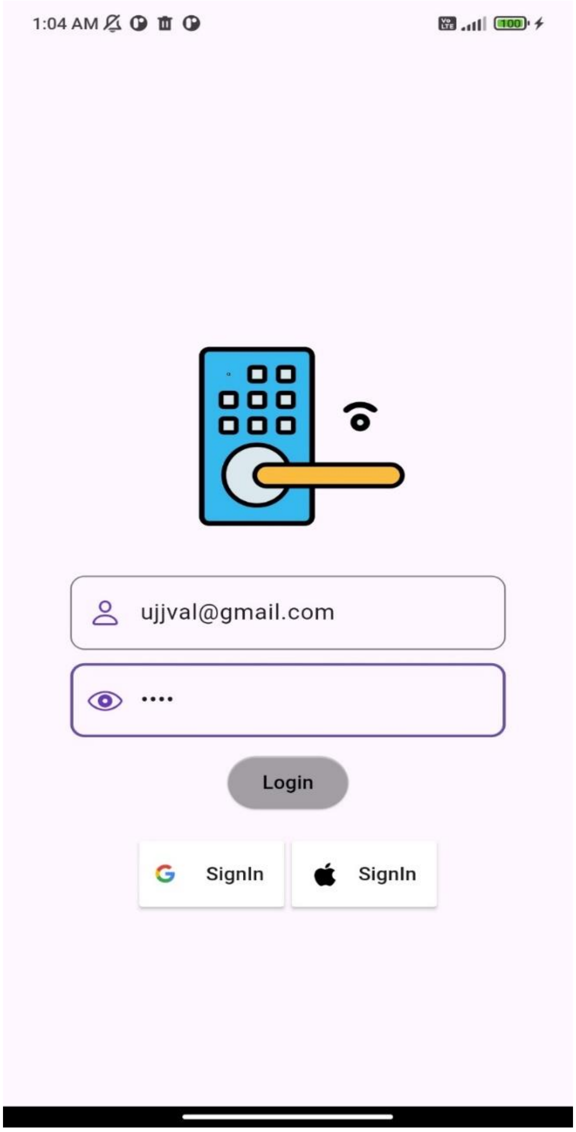


Screen Layout





Invalid password. Please try again.



1:04 AM

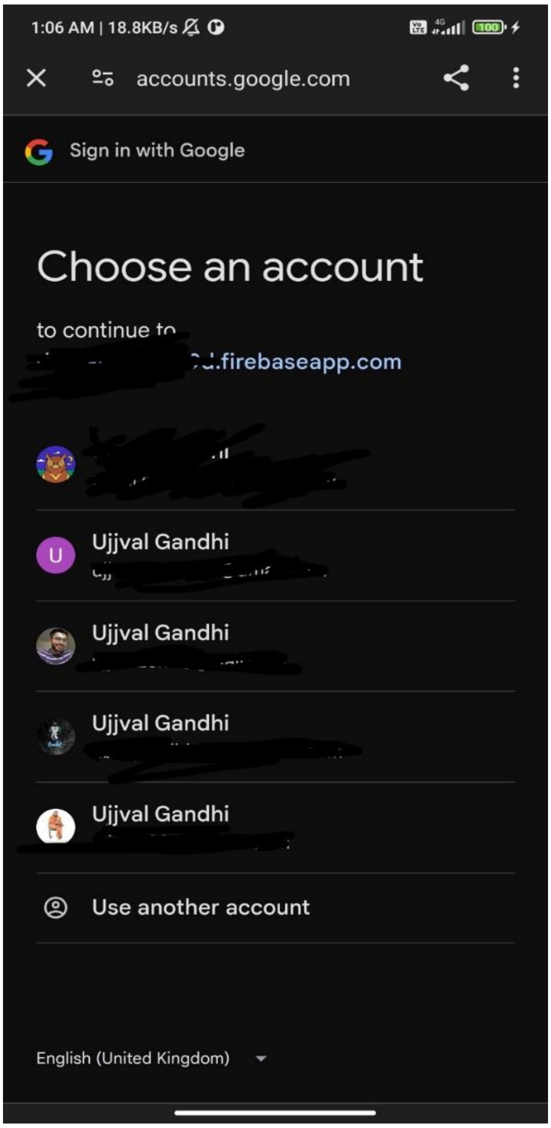
ujjval@gmail.com

••••

Login

SignIn

SignIn



1:06 AM | 18.8KB/s

accounts.google.com

Sign in with Google

Choose an account

to continue to [redacted].firebaseapp.com

[redacted]

Ujjval Gandhi
[redacted]

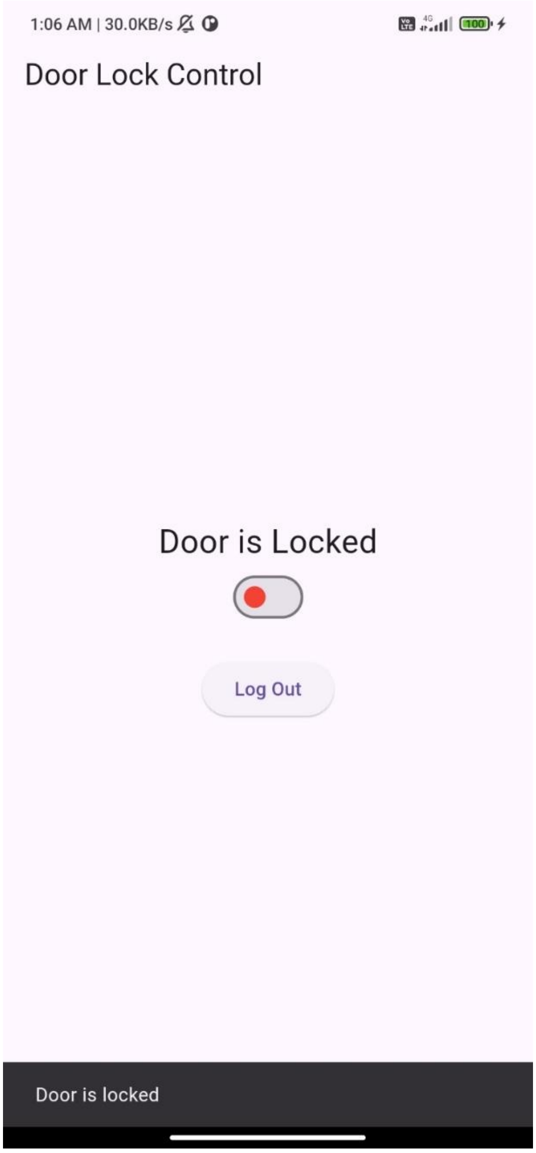
Ujjval Gandhi
[redacted]

Ujjval Gandhi
[redacted]

Ujjval Gandhi
[redacted]

Use another account

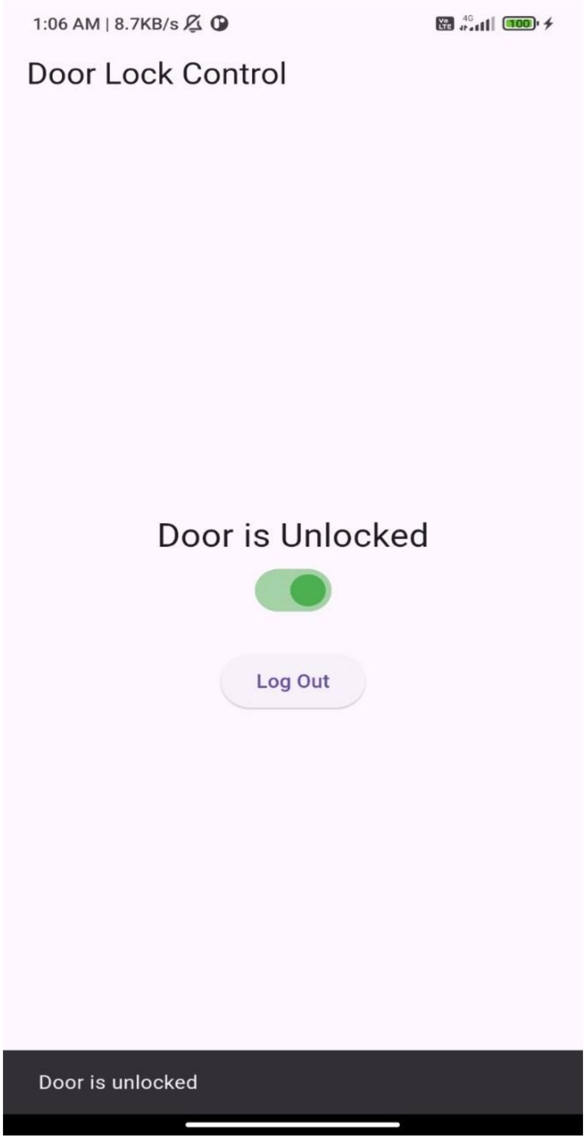
English (United Kingdom)



Door Lock Control

Door is Locked

Door is locked



Door Lock Control

Door is Unlocked

Door is unlocked

SYSTEM STUDY

➢ **Existing Study :**

Currently it is observe that door lock is controlled with hardware devices like Arduino, ESP 32 , and more that can easily control door lock with their respective functionalities.

➢ **Proposed System :**

Though the rate of success is less in terms of functionalities and people are still in hope that there will be door lock with all functionalities in a single lock. In our door lock, we simply tried to control it with Wifi via device (Cell Phone).

➢ **Scope of the proposed System :**

To be frank with the scope of our system, it is a door lock that has a functionality that is, it can be connected with Wifi and easily controlled. This project can help in several unique way and dealing several day-to-day tasks.

➢ **Aim and Objective proposed System :**

The aim is to develop a secure IoT-based smart door lock system using RFID, fingerprint recognition, and a Flutter app with MQTT for remote access. The system ensures enhanced security through multi-factor authentication, real-time monitoring, and remote control. It provides a user-friendly interface for managing access permissions and tracking activity logs. The design emphasizes scalability, low power consumption, and secure data handling for reliable performance.

➢ What we actually did in the project?

- ✓ Firstly, we have gathered all the above-mentioned tools and install the Arduino IDE.
- ✓ Then we had completed door lock with the help of fingerprint, RFID and power supply by fitting them all via wires.
- ✓ After the completion of it, we tested Arduino board and other components, about the condition of their working.
- ✓ Thereafter, we connected Arduino Board with ESP32 Wifi module and Fingerprint Senser. Similarly, I2C LCD was connected with Arduino Uno and 12-volt batteries for power supply.
- ✓ Once the connections were made, we verified code and uploaded the code on Arduino Uno Board in Arduino IDE software by selecting board and respective port via power cable of Arduino Uno.
- ✓ It was finally followed by application on cell phone that helped us to connect with ESP32 Wifi module that made door lock working.

➢ Feasibility Study:

- ✓ Feasibility study is high-level capsule version of entire system analysis and design process.
- ✓ The study begins by classifying the problem definition.
- ✓ Feasibility study tried to determine whatever a given. solution would work or not.
- ✓ Its main objective is not to solve the problems, but to acquire its scope.
- ✓ It focuses on the following stated parameters:
    - ▪ Meet user requirements
    - ▪ Best utilization of available resources
    - ▪ Develop a cost-effective system

A. OPERATIONAL FEASIBILITY :

- The door lock easy to operate as it is directly connected to cell phone with the medium of Wifi.

B. TECHNICAL FEASIBILITY :

- ❖ Hardware needed
    - Arduino Uno
    - ESP32
    - Batteries and cables
    - Laptop or personal computer.

- ❖ Software required
    - Arduino IDE
    - Android Studio

C. ECONOMICAL FEASIBILITY :

- As there is specific hardware requirement in this project which are individually low in cost and are easily available in shops or online mode. As far as software cost is concerned, they are open source so the product is economically feasible.

SYSTEM TESTING

- While working on project, hardware all the hardware were tested prior to their usage.

- Before Uploading code, code was verified and detection of board and port was checked.

- Connectivity of Bluetooth module HC-05 with cell phone was tested and then executed.

FUTURE ENHANCEMENT

Future enhancements for the smart door lock system include integrating voice assistant compatibility, facial recognition, and geofencing for added convenience. Features like temporary access codes, tamper alerts, and cloud-based logs can enhance security and usability. Additionally, smart home integration and energy-efficient designs can improve system functionality and user experience.

REFERENCES

https://randomnerdtutorials.com/

https://pub.dev/

https://testclient-cloud.mqtt.cool/

https://www.tinkercad.com/

https://www.youtube.com/

https://console.firebase.google.com/u/0/