NEW HORIZON
COLLEGE OF ENGINEERING

PROJECT REPORT

ON

"No Key – Smart Unlock Using Fingerprint"

Submitted in the partial fulfilment of the requirements in the 8th semester of

BACHELOR OF ENGINEERING
IN
INFORMATION SCIENCE AND ENGINEERING

By

GanaSasank Reddy [1NH19IS195]
Hast Sinha [1NH19IS194]
Harsh Vardhan Jaiswal [1NH19IS055]
Aditya Sunit Kanoi [1NH19IS009]

For
COURSE NAME : PROJECT PHASE 2
COURSE CODE : 20ISE83A

Under the guidance of,

Dr. B Swathi
Associate Professor , Dept. of ISE, NHCE

ACADEMIC YEAR : 2022-2

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **NO KEY- Smart Door Unlock System Using Fingerprint** carried out by GanaSasank Reddy, Harsh Vardhan Jaiswal , Aditya Sunit Kanoi and Hast Sinha, bearing USN 1NH19IS195, 1NH19IS055, 1NH19IS009 and 1NH19IS194, Bonafede students of 8th semester in partial fulfilment for the award of Bachelor of Engineering in Information Science & Engineering of the Visvesvaraya Technological University, Belagavi during the year 2022-23. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library.

The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said Degree.

Name & Signature of the Guide	Name & Signature of the HOD
Dr. B Swathi	**Dr. Mohan H S**

External Viva

Name of the Examiners	Signature with Date
1. ……	……..……..
2. ….……...	…….....

NEW HORIZON COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that we have followed the guidelines provided by the Institution in preparing the project report and presented report of project titled "No Key – Smart Door Unlock System Fingerprint" and is uniquely prepared byus after the completion of the project work. We also confirm that the reportis only prepared for my academic requirement and the results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

Signature of the Students

Name: Aditya Sunit Kanoi	**Name: Hast Sinha**	**Name: GanaSasank Reddy**	**Name: HarshVardhan Kumar Jaiswal**
USN: 1NH19IS009	**USN: 1NH19IS194**	**USN: 1NH19IS195**	**USN: 1NH19IS055**

ABSTRACT

Technology has improved, and smart locking systems have become more sophisticated. In this case, the android-based Smart System is primarily intended for multimode operations. Such a system is necessary for banks and businesses since it provides functions that let users control locks.

The implementation efficiency of the system is incredibly helpful because of its functionality and user-friendly interface. Some homeowners aim to connect their home's numerous home automation devices. Those connected to a Windows-based PC are the most popular home controllers. In our study, we introduced a form of smart technology that utilizes Bluetooth while using a mobile smartphone. Consequently, using it will be simpler and more effective. Additionally, it supported the free and open-source Android and Arduino platforms. This paper proposes a door lock automation system that uses an Android smartphone with Bluetooth as the first piece of hardware. Following a description of the design and software development process, a Bluetooth-based Smartphone application for locking and unlocking doors is demonstrated. The task module acts as the agent in the hardware design for the door-lock system, the Arduino microcontroller serves as the controller and data processing hub, and the solenoid acts as the door-lock output. The results of each test show that it is compatible with the original plan for this study.

ACKNOWLEDGEMENT

Any project is a task of great enormity, and it cannot be accomplished by an individual without support and guidance. I am grateful to a number of individuals whose professional guidance and encouragement have made this project completion a reality.

I have a great pleasure in expressing my deep sense of gratitude to the beloved Chairman **Dr. Mohan Manghnani** for having provided me with a great infrastructure and well-furnished labs.

I take this opportunity to express my profound gratitude to the Principal **Dr. Manjunatha** for his constant support and management.

I am grateful to **Dr. Mohan H.S.**, Professor and Head of the Department of ISE, New Horizon College of Engineering, Bengaluru for her strong enforcement of perfection and quality during the course of my project work.

I would like to express my thanks to the guide **Dr. B. Swathi**, Associate Professor, Department of ISE, New Horizon College of Engineering, Bengaluru who has always guided me in detailed technical aspects throughout my project.

I would like to mention special thanks to all the Teaching and Non-Teaching staff members of Information Science and Engineering Department, New Horizon College of Engineering,Bengaluru for their invaluable support and guidance.

GanaSasank Reddy || (1NH19IS195)
HarshVardhan Jaiswal || (1NH19IS055)
Aditya Sunit Kanoi || (1NH19IS009)
Hast Sinha || (1NH19IS194)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

PREAMBLE

Locks are very important security in daily life so we should be very careful while choosing locks that's why we are creating a secured smart fingerprint-based lock system.

1.1 Introduction

Despite the abundant benefits of biometrics-based private confirmation structures over traditional freedom orders established tokens or information, they are susceptible to attacks that can decrease their protection considerably. Biometrics-located individual authentication scheme that uses corporeal (mark on finger, face) or behavioral (talk, manuscript) characteristics are becoming more and more well-known, distinguished to traditional holes that are established tokens (key) or information(password).

1.2 The Motivation for the Project

Most doors in the current system are managed by individuals using Keys, security cards, and a password to enter locked doors. The development of wireless controlling technology is a result of this. With a skillful fusion of modern technology and embedded systems, the purpose is achieved. However, we're planning to make a modification that will make things more secure when compared to the current system the existing system used regular, conventional locks that have been around for quite previous time Traditional locks do have a drawback, though, in that they may be readily broken by burglars, or they can go missing. Then we upgraded to modern locks that can be used with pin numbers, passwords, and other security measures.

1.3 Problem Statement

Fingerprint-located labelling is individual of the most main biometric sciences that have drawn a solid amount of consideration currently. Fingerprint technology is so low in individual identification that it has existed well settled. Each human has a singular owns the mark, even the twin has various marks on the finger. So, fingerprint acknowledgment is a valuable instability law request. The photoelectric lock utilizing fingerprint acknowledgment includes a process of proving the user's similarity by utilizing mark-on-finger recognition as a key to the photoelectric lock. Then

the facts of the recognized dab countenance will be stored in a table for verification and approval by the consumer. These marks on finger recognition orders establish the theory that the human fingerprint is singular. It is mainly to confirm the individuality of dab in consideration of using the dab image for protection.

1.4 Existing System

The earliest patent for a double-acting pin tumbler lock was granted to American physician Abraham O. Stansbury in England in 1805, but the modern version, still in use today, was invented by American Linus Yale Sr. in 1848. This lock design used pins of varying lengths to prevent the lock from opening without the correct key. In 1861, Linus Yale Jr. was inspired by the original 1840s pin-tumbler lock designed by his father, thus inventing and patenting a smaller flat key with serrated edges as well as pins of varying lengths within the lock itself, the same design of the pin-tumbler lock which remains in use today.

The modern Yale lock is essentially a more developed version of the Egyptian lock. An electromechanical device known as a relay can be used to create or sabotage an electrical connection. In essence, a relay is just like a mechanical change, except that you may control it with a digital sign rather than alternately turning it.

Then we upgraded to modern locks that can be used with pin numbers, passwords, and other security measures. The drawback of these locks is that we can forget our passwords or pin numbers. We also have biometrics that can be used to open the door lock itself, but we are now proposing a new style of modern wireless door lock utilizing the same biometrics.

1. VIZpin's Entegrity™ Smart Lock

Discrete, low-profile exterior matches any décor, unique Bluetooth design with no network connection ,4 AA batteries Operated, easy installation , manage access anytime, from anywhere, using simple VIZpin cloud tools , perfect for apartments, offices, equipment rooms, restrooms .

1.1 Limitations of Existing Systems

You may be the one to forget your keys now and then, and it can be easy to forget you PIN code for the lock and when you're in a rush to get into the room or building or it is nighttime and dark, you don't want to be changing the code in the middle of the night or when it's raining.

Only tell the code to people who you trust, as you don't want a code to your property to be local news. When the lock has been used a few too many times, the coating may start to come away or mucky fingerprints may start to occur on the buttons. Keep the lock maintained and clean to stop unwanted people finding out the code.

Some digital door locks have a PIN code length of up to 10 digits - this is not what you want! Digital Door Locks will be much more secure if they are only 4 digits long. Purchase a quality lock that you can change the PIN code on, don't buy locks that are provided with a PIN code because people can find out the code.

1.5 Proposed System

We are developing a door lock that can be opened using biometrics in the proposed system. These days, it is fashionable for them to be secure and simple for the owner or administrator to open. We used a biometric door lock to create this, however, a cell phone will be used to unlock the lock.

We are developing an app that will enable mobile phone door lock access. Can be connected through Bluetooth. To build an app, we're using Kotlin, and we've given Bluetooth access to operate the door lock. Using Android Studio, we are developing a kotlin application in which we will use the Java programming language. Kotlin was used to create the code to communicate with the HC-05 Bluetooth module.

To connect with the HC-05 module, a basic Bluetooth adapter and a Bluetooth socket (integrated within the software) were utilized. Additionally, this connection allowed data to be sent across the socket from the Rx and Tx pins of the Arduino Nano. The software instantly establishes a connection with the HC-05 Board, and the device's fingerprint scanner serves as an authentication factor.

Through Arduino, we must provide a power source with an appropriate voltage. Open the app that was developed using the Kotlin application after providing electricity. The mobile device must be turned on for Bluetooth and connected to the HC-05 Bluetooth Module before the Bluetooth symbol in an app change to a lock icon. Next, we must tap the fingerprint symbol. It will notify us when we touch it that a fingerprint is required to unlock it. Keep your finger on the fingerprint reader on your phone right now. If the fingerprint is recognized by our phone, the lock is turned on and the lock icon is changed to an unlock icon.

A Bluetooth module called HC-05 is created for wireless communication. This module may be set up as either a grasp or a slave. The HC-05 contains a red LED that indicates the connection status, including whether Bluetooth is involved. Before being connected to the HC-05 module, this red LED constantly pulses periodically.

Its blinking reduces to two seconds when it is linked to another Bluetooth device. 3.3 V is needed for this module. We can also connect a 5V supply voltage because the module includes built-in 5 to 3.3 V regulators. There is no need to change the HC-05 Bluetooth module's transmit level because it contains a V stage for RX/TX and the microcontroller can sense 3.3 V levels.

1.6 Objective

- Unique Bluetooth design with no network connection.
- Secure Arduino Based Chipsets.
- No Physical Key Requirement.
- Hard to Replicate.
- Nearly Unbackable and Traceable.
- Long Distance Auto Connection Mechanism.
- Fast, Secure and Modern Day Tool.

CHAPTER 2

LITERATURE SURVEY

This chapter provides an insight into a few of the papers which were taken as reference for the implementation of the project.

2.1 REFERENCE PAPER 1

TITLE: SMART DOOR UNLOCK SYSTEM USING FINGERPRINT

AUTHORS: K.Rajesh , B.Venkata Rao,P.AV.S.K.Chaitanya ,A.ruchitha Reddy ,Department of ECE, Chalapathi Institute of Engineering and Technology, Lam, Guntur, AP, India -522034

ABSTRACT:

Human detection and recognition field is very significant and has undergone rapid change with time. An important and very reliable human identification method is fingerprint identification. Fingerprint of every person is unique. So, this helps in identifying a person or in improving security of a system. Fingerprint of a person is read by a special type of sensor. Fingerprint sensor can be interfaced with a microcontroller. Through keypad we can add new user and delete the existing user, also identify the user by selecting corresponding option through keypad. In this project we use a fingerprint sensor to read one's identity to automatically operate the door of the car. For this, we use a microcontroller to enable the door opening or closing if the matching between scanned data and the already existing data is correct. Comparison is done inside the fingerprint module itself and its output is given to microcontroller. Result is displayed in a LCD display whether the user is authorized or not. LCD also helps to make troubleshooting easier. Alarming option is provided to warn about an unauthorized usage. Microcontroller used is PIC16F877.

2.2 REFERENCE PAPER 2

TITLE: SMART LOCKS: EXPLORING SECURITY BREACHES AND ACCESS EXTENSIONS

AUTHORS: PALLE, SAIPRASANNA ,Bachelor of Technology in Computer Science Jawaharlal Nehru Technological University, Hyderabad, Telangana, 2012

ABSTRACT:

The Internet of Things (IoT) has rapidly become one of the most popular devices across businesses and technology. Referred to as the next industrial revolution, it has transformed the way devices interact with each other, with home automation being the one of the popular fields of IoT. Though IoT can make the devices smarter, it also has a potential to expose these same devices to an attacker, to exploit their vulnerabilities, thus raising a concern for security. This paper deals with the security aspects of one of the popular subsets of home automation systems: smart locks. Smart locks replace the traditional door locks; they can be electronically controlled by mobile devices or the lock manufacturer's remote servers. The current state of art, design, implementation, vulnerabilities and the attacks that can be exploited on these devices are discussed. A solution that addresses the design and architecture of IoT smart lock in providing a better security mechanism is proposed for achieving smarter and secure smart homes. It is shown through experimental analysis that the proposed smart lock system is secure against unauthorized access, replay, and relay attacks.

2.3 REFERENCE PAPER 3

TITLE: Smart Door Locking System

AUTHORS: Nishad N. Gupte, Mihir R. Shelar , Department of Electronics Engineering, Datta Meghe College of Engineering, Airoli.

ABSTRACT:

Today's world is a smart world we live in, and Smart Phones have a major contribution in it. A smart phone is now a very common device that everyone carries with them all the time. Smart phones have enabled us to perform various tasks while using a single device. Considering this phenomenon, we can exploit the idea of remote controlling a door with the help of a smart phone. Physical keys and a lock are the basic requirements for a door. However, managing these keys has become cumbersome. To overcome this, we present a solution which is a smart and secure way of remote controlling the door using a smart phone.

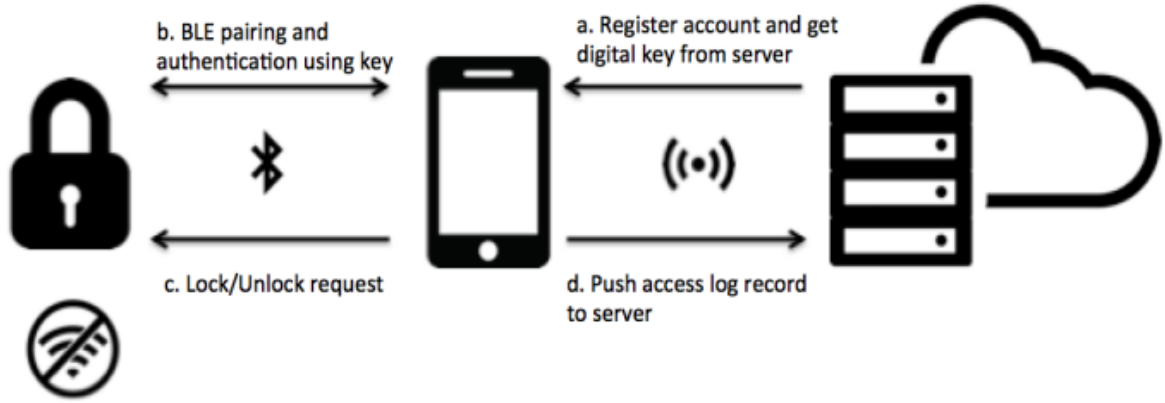


Figure 2.1: Device-Gateway-Cloud Architecture.

2.4 REFERENCE PAPER 4

TITLE: Smart Door Lock

AUTHORS: Worcester polytechnic institute - Aleksander ibro, Augusto wong , Mario zyla.

ABSTRACT:

The goals of this project were to build a modern, easy-to-use, smart door lock that allows for accessible unlocking and adds convenience, utility, and security to your home. It allows users to open their door remotely via the accompanying Smart Lock mobile app, or hands-free by using face recognition via a camera mounted on the door. The system was made up of three major components, including a cloud back-end, an on-board logical unit, and a mobile application.

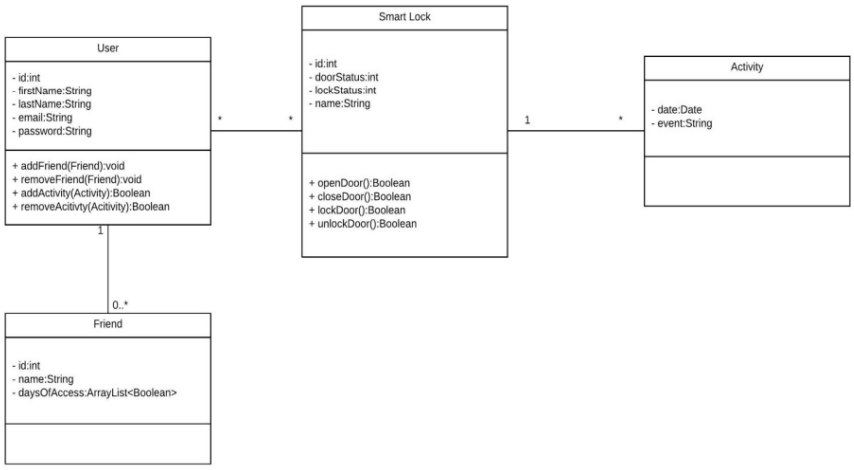


Figure 2.2: High-level class diagram of the Smart Lock Main Objects.

2.5 REFERENCE PAPER 5

TITLE: Programming IoT Devices by Demonstration Using Mobile Apps

AUTHORS: Toby Jia-Jun Li1, Yuanchun Li , Fanglin Chen , and Brad A. Myers , Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA.

ABSTRACT:

The revolutionary advances of Internet of Things (IoT) devices and applications have helped IoT emerge as an increasingly important domain for end-user development (EUD). Past research has shown that end users desire to create various customized automations, which would often utilize multiple IoT devices. Many solutions exist to support EUD across multiple IoT devices, but they are limited to devices from the same manufacturer, within the same "ecosystem" or supporting a common API. We present EPIDOSITE, a mobile program- ming-by-demonstration system that addresses this limitation by leveraging the smartphone as a hub for IoT automation. It enables the creation of automations for most consumer IoT devices on smartphones by demonstrating the desired behaviors through directly manipulating the corresponding smartphone app for each IoT device. EPIDOSITE also supports using the smartphone app usage context and external web services as triggers and data for automations, enabling the crea- tion of highly context-aware IoT applications.

Figure 2.3: The architecture of the EPIDOSITE external service trigger mechanism

CHAPTER 3

PROPOSED METHODOLOGY

The proposed methodology for the project is discussed in this chapter.

3.1 Required Components and their Descriptions

Figure 3.1: Block Diagram of Required Components

1. Arduino Nano
2. Bluetooth module HC-05
3. The Solenoid locks
4. Relay module
5. Connecting Wires
6. Battery

1. A Bluetooth module called HC-05 is created for wireless communication. This module may be set up as either a grasp or a slave. The HC-05 contains a red LED that indicates the connection status, including whether Bluetooth is involved. Before being connected to the HC-05 module, this red LED constantly pulses periodically.

2. Arduino nano is an open-source tool for prototyping electronic devices that relies on flexible, easy-to-use hardware and software. For artisans, architects, and professionals, and everyone was fascinated by creating clever articles or situations. It's a large figuring step based on an open source

microcontroller board, and advancement environment for writing board programming programmers. Simply put I n other terms, Arduino is a tiny microcontroller board with a USB connector to connect to your computer and some networking add-ons that could be connected to external devices, such as engines, transfers, light sensors, laser diodes, amplifiers, mouthpieces, etc.

Table 3.1: Arduino Nano Model B Features

Processor	Broadcom 2711, Quad-core Cortex A72 64-bit SoC @ 1.5GHz
Memory	1GB, 2GB or 4GB LPDDR4 SDRAM
Connectivity	2.4GHz/5.0GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 5.0, BLE 2 x USB 2.0/2 x USB 3.0 ports delivering true Gigabit Ethernet
Access	Extended 40-pin GPIO header
Video and Sound	2 x micro-HDMI, 4k video 1 x MIPI DSI display port 1 x MIPI CSI camera port 4 pole stereo output and composite video port
Multimedia	H.265 decode (4kp60) H.264 decode (1080p60) H.264 encode (1080p30) OpenGL ES 1.1, 2.0, 3.0 graphics
SD Card Support	Micro SD format for loading OS & data storage
Input Power	5V/3A DC via USB type C connector SV DC via GPIO POE enabled

3. It is available in opening in the force-on mode type as well as bolting and protecting in this type, which may be used individually for circumstances. Only when the solenoid is powered on is opening possible with the strength-on opening type. When an entrance of this kind is guarded, it won't open in the event of an energy outage or twine separation, providing excellent assurance. This kind is used specifically in locations where it is necessary to prevent misconduct.

4. An electromechanical device known as a relay can be used to create or sabotage an electrical connection. In essence, a relay is just like a mechanical change, except that you may control it with a digital sign rather than alternately turning it on and off. It consists of a flexible transferable mechanical phase that can be managed electronically using an electromagnet.

5. In contemporary connections and energy links, artificial polymer protection that resembles elastic is employed. Because of its chief moisture hindrance, it was built underground. The most severe working temperature at the conductor surface and the recommended working voltage are used to evaluate protected connections.

6. A battery is a tool that is constructed of one or more electrochemical cells with external connections and is used to power electrical devices including mobile phones, flashlights, and electric vehicles.

7. Android Studio

Android Studio is an IDE used to develop Mobile Applications. Android Studio adds new capabilities to help us be more productive while developing Android apps. Android Studio will be utilized to build our Mobile Application.

3.2 Proposed Implementation

We are developing an app that will enable mobile phone door lock access. Can be connected through Bluetooth and can be easily unlocked and locked with fingerprint sensor of the Android , IOS based Smartphones.

To build an app, we're using Kotlin, and we've given Bluetooth access to operate the door lock. Using Android Studio, we are developing a kotlin application in which we will use the Java programming language. Kotlin was used to create the code to communicate with the [10-13] HC-05 Bluetooth module.

1. The Creation of the Digital Encrypted hardware lock.

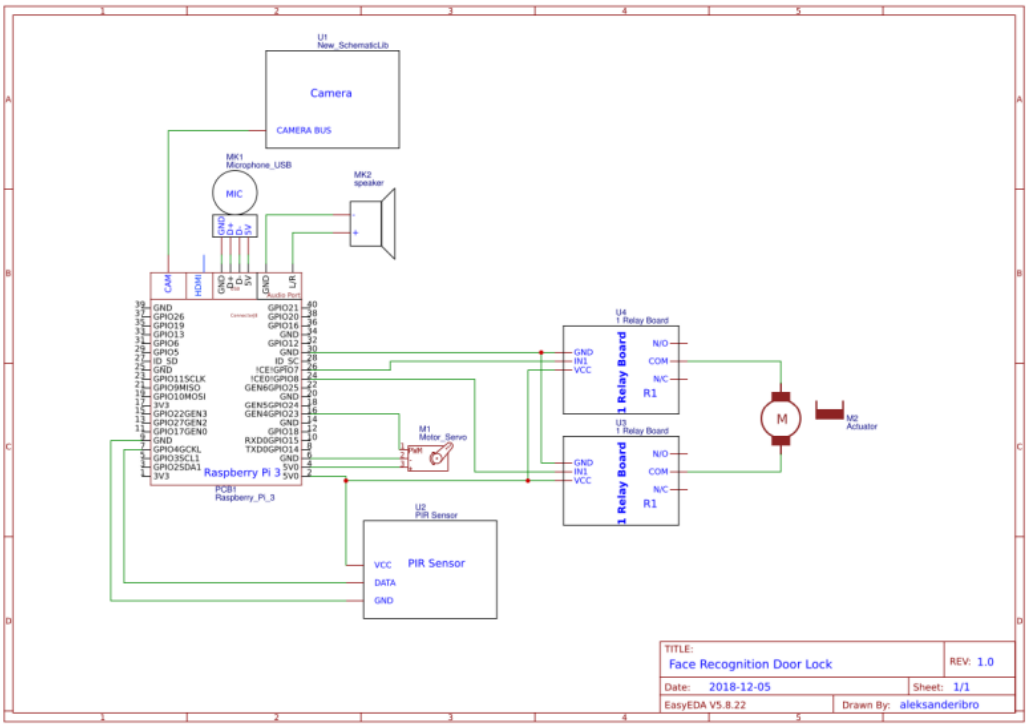


Figure 3.2 Electrical Schematic of the Smart Door Lock device.

2. Mobile Application

App Services is an Azure tool that offers a backend for mobile apps, allowing quick development and deployment [3]. Initially, our cloud backend was provided by the Flask server, over HTTP requests to the SQL database. We realized this was very slow and not secure, as it was over HTTP, which is why we migrated to Azure App Services. This allowed a quick connection via a token from the app to the SQL DB, thus avoiding request handling by the Flask server. The difference in the requests between requests being handled by the Flask server and requests being handled by App Services.

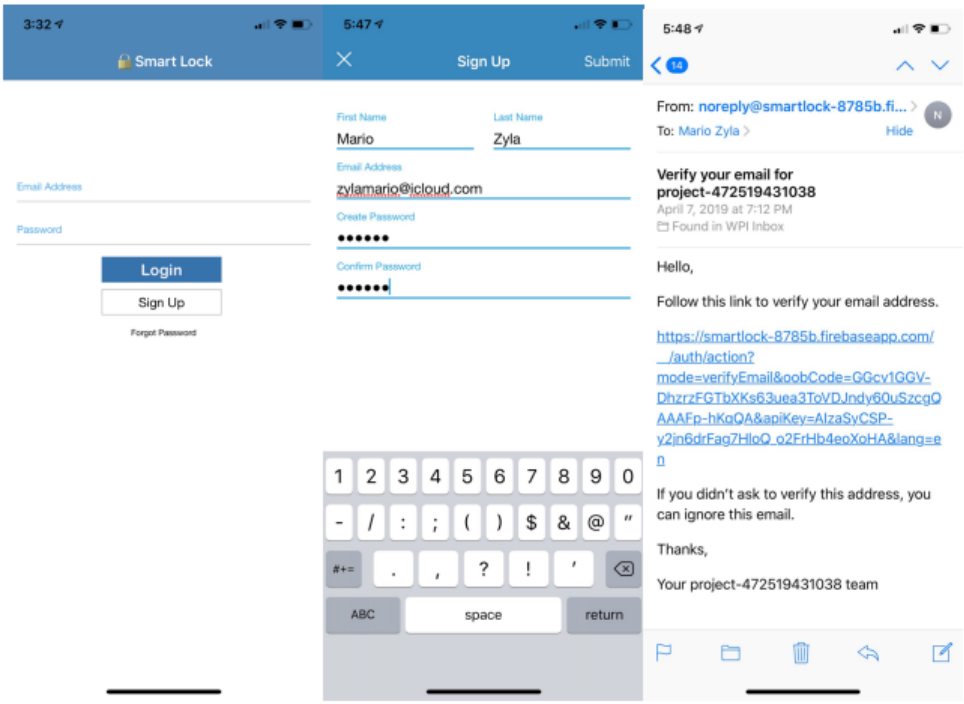


Figure 3.3: Expected Mobile Application

3.3 Expected Results

The program for ARDUINO NANO was written in C++ language. It was then compiled using ARDUINO IDE. After that code is dumped or uploaded into a microcontroller. We have created android app by using KOTLIN Java Language. After that we are connecting the hardware after giving power supply. The android app communicating with ARDUINO NANO using wireless Bluetooth communication. If the communication happens in a correct way, the door lock opens and closes. The communication and result flow below using flow chart.

1. IoT System Embedded on Physical Locks.
2. Android Application for working and user.
3. Fingerprint based lock and unlock mechanism.
4. Bluetooth based connectivity of lock system.

Figure 3.4: Smart Door Lock and Android Application

Our team successfully implemented a Smart Door Lock system with a seamless integration of a cloud backend, an on-board logical unit like the Raspberry Pi, and a user-friendly mobile application. All the features initially proposed, except for a Facebook login, were implemented and tested, including some extra ones, as explained in Section 3.6. We made sure that the requests and users' information is secure by using Microsoft's Cloud and their App Services tool. We were able to build a mockup as a door system, using some wooden platforms and WPI's 3D printers. Figure above shows an image of the finished Smart Lock system mounted on the mentioned platform.

CHAPTER 4

HARDWARE AND SOFTWARE SPECIFICATIONS

Coding Implementations:

Android Manifest File :

The AndroidManifest.xml file is an essential part of an Android application. It contains critical information about the app, such as its package name, version code, permissions, activities, services, broadcast receivers, and content providers. The Android system reads this file to learn about the app's components and their capabilities before launching the app.

Here are some of the essential elements of the AndroidManifest.xml file and their functions:

1. Package Name: The package name is a unique identifier for the app. It helps the Android system distinguish the app from other installed apps. The package name is defined in the manifest file and should be unique for every app.
2. Activities: Activities are the building blocks of an Android app. They represent the individual screens or UI components of an app. The manifest file declares all the activities that an app has. Activities are also declared in the manifest file with a unique name, a label for the user, and an intent filter.
3. Intent Filters: Intent filters define the types of intents that an activity can handle. They are defined in the manifest file using action, category, and data tags. The intent filter is used to declare how your activity can be launched. For example, if you want your activity to be launched when the user clicks on a particular link, you can define an intent filter that specifies the type of link your activity can handle.
4. Permissions: Permissions are rules that an app must follow to access protected system resources or sensitive user data. These permissions must be declared in the manifest file. If an app tries to access a resource that requires a permission, the system will prompt the user to grant or deny the permission.
5. Services: Services are used to perform background tasks in an app. The manifest file declares all the services that an app has. Services are also declared in the manifest file with a unique name, a label for the user, and an intent filter.
6. Broadcast Receivers: Broadcast Receivers are used to listen to system-wide broadcast events or messages and respond to them. For example, an app might use a broadcast receiver to listen for an incoming SMS message. The manifest file declares all the broadcast receivers that an app has. Broadcast receivers are also declared in the manifest file with a unique name and an intent filter.

In summary, the AndroidManifest.xml file is a vital part of an Android app that defines essential information about the app's components, capabilities, and permissions. It helps the Android system understand the app and launch it correctly.

CODE :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1"
    android:versionName="1.0"
    android:compileSdkVersion="31"
    android:compileSdkVersionCodename="12"
    package="com.  teamhash.arduino"
    platformBuildVersionCode="31"
    platformBuildVersionName="12">

    <uses-sdk
        android:minSdkVersion="21"
        android:targetSdkVersion="31" />

    <uses-permission
        android:name="android.permission.BLUETOOTH" />

    <uses-permission
        android:name="android.permission.BLUETOOTH_ADMIN" />

    <uses-permission
        android:name="android.permission.BLUETOOTH_CONNECT" />

    <uses-permission
        android:name="android.permission.USE_BIOMETRIC" />

    <uses-permission
        android:name="android.permission.USE_FINGERPRINT" />

    <application
        android:theme="@ref/0x7f11019b"
        android:label="FingerPrintLocker"
```

```
        android:icon="@ref/0x7f080099"
        android:debuggable="true"
        android:testOnly="true"
        android:allowBackup="true"
        android:supportsRtl="true"
        android:appComponentFactory="androidx.core.app.CoreComponentFactory">

        <activity
            android:name="com.  teamhash.arduino.MainActivity"
            android:exported="true">

            <intent-filter>
                <action
                    android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <provider
            android:name="androidx.startup.InitializationProvider"
            android:exported="false"
            android:authorities="com.  teamhash.arduino.androidx-startup">

            <meta-data
                android:name="androidx.emoji2.text.EmojiCompatInitializer"
                android:value="androidx.startup" />

            <meta-data
                android:name="androidx.lifecycle.ProcessLifecycleInitializer"
                android:value="androidx.startup" />
        </provider>
    </application>
</manifest>
```

XML Layouts :

XML Layouts play a critical role in Android application development. They define the user interface of the application and how the elements should be positioned and interact with each other. Here are some of the key aspects of XML Layouts:

1. UI definition: XML Layouts provide a way to define the user interface of an Android application. You can define the visual appearance of each screen or activity in the app, including text, images, buttons, and other UI elements.
2. Flexibility: XML Layouts allow you to create layouts that are flexible and can adapt to different screen sizes and orientations. You can use different layout types such as LinearLayout, RelativeLayout, and ConstraintLayout to achieve the desired UI design.
3. Separation of concerns: XML Layouts promote the separation of concerns between the UI and the underlying business logic of the application. By keeping the UI definition separate from the code that handles the business logic, you can make changes to the UI without affecting the functionality of the app.
4. Reusability: You can reuse XML Layouts across multiple activities or screens in the same application or even across different applications. This reduces development time and effort as you do not have to create the same UI elements from scratch every time.
5. Accessibility: XML Layouts enable you to create an accessible user interface for your app. You can use attributes such as content description, focus order, and view hierarchy to make your app accessible to users with disabilities.

Overall, XML Layouts are a critical part of Android application development as they define the visual appearance and behavior of the app's user interface.

CODE :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button_lock"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lock"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/button_unlock"
        app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent" />

  <Button
    android:id="@+id/button_unlock"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Unlock"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/button_lock"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Main Activity :

In Android development, the main activity is the entry point for an application. It is the first screen that appears when a user launches the application. The main activity sets up the initial user interface and provides a way for the user to interact with the application.

The main activity is defined in the AndroidManifest.xml file and is typically written in Java or Kotlin. When the user launches the application, the Android operating system starts the main activity, which then creates and displays the initial user interface.

The main activity typically contains a layout file, which describes the user interface. The layout file is written in XML and defines the position, size, and appearance of the various user interface elements, such as buttons, text fields, and images.

The main activity also contains the logic that controls how the user interacts with the application. This logic is implemented in the form of event handlers that respond to user actions, such as button clicks or text input.

In addition, the main activity may communicate with other parts of the application, such as other activities or services, in order to retrieve data or perform background tasks.

Overall, the main activity is a crucial component of an Android application, as it provides the initial user interface and handles user interactions.

CODE:

```
import android.bluetooth.BluetoothAdapter
import android.bluetooth.BluetoothDevice
import android.bluetooth.BluetoothSocket
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import java.io.IOException
import java.util.*

class MainActivity : AppCompatActivity() {

    // Define the UUID for the Bluetooth service
    private val SERVICE_UUID: UUID = UUID.fromString("00001101-0000-1000-8000-
00805F9B34FB")

    // Define the MAC address of the Bluetooth module on the Arduino board
    private val DEVICE_ADDRESS: String = "00:00:00:00:00:00"

    // Define the BluetoothAdapter, BluetoothDevice, and BluetoothSocket
    private var bluetoothAdapter: BluetoothAdapter? = null
    private var bluetoothDevice: BluetoothDevice? = null
    private var bluetoothSocket: BluetoothSocket? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Initialize the BluetoothAdapter
        bluetoothAdapter = BluetoothAdapter.getDefaultAdapter()

        // Check if Bluetooth is available on the device
```

```
        if (bluetoothAdapter == null) {
            Toast.makeText(this, "Bluetooth is not available", Toast.LENGTH_LONG).show()
            finish()
        }

        // Check if Bluetooth is enabled on the device
        if (!bluetoothAdapter!!.isEnabled) {
            Toast.makeText(this, "Please enable Bluetooth", Toast.LENGTH_LONG).show()
            finish()
        }

        // Get the BluetoothDevice for the specified MAC address
        bluetoothDevice = bluetoothAdapter!!.getRemoteDevice(DEVICE_ADDRESS)

        // Create a BluetoothSocket for the BluetoothDevice
        try {
            bluetoothSocket =
bluetoothDevice!!.createRfcommSocketToServiceRecord(SERVICE_UUID)
        } catch (e: IOException) {
            Log.e("Bluetooth", "Error creating Bluetooth socket", e)
            Toast.makeText(this, "Error creating Bluetooth socket", Toast.LENGTH_LONG).show()
            finish()
        }

        // Connect to the BluetoothSocket
        try {
            bluetoothSocket!!.connect()
        } catch (e: IOException) {
            Log.e("Bluetooth", "Error connecting to Bluetooth socket", e)
            Toast.makeText(this, "Error connecting to Bluetooth socket",
Toast.LENGTH_LONG).show()
            finish()
        }
```

```
        // Get references to the lock and unlock buttons
        val lockButton = findViewById<Button>(R.id.lock_button)
        val unlockButton = findViewById<Button>(R.id.unlock_button)

        // Set click listeners for the lock and unlock buttons
        lockButton.setOnClickListener {
            // Send "lock" command to the Arduino board
            try {
                bluetoothSocket!!.outputStream.write("lock".toByteArray())
            } catch (e: IOException) {
                Log.e("Bluetooth", "Error sending command to Bluetooth device", e)
                Toast.makeText(this, "Error sending command to Bluetooth device",
Toast.LENGTH_LONG).show()
            }
        }

        unlockButton.setOnClickListener {
            // Send "unlock" command to the Arduino board
            try {
                bluetoothSocket!!.outputStream.write("unlock".toByteArray())
            } catch (e: IOException) {
                Log.e("Bluetooth", "Error sending command to Bluetooth device", e)
                Toast.makeText(this, "Error sending command to Bluetooth device",
Toast.LENGTH_LONG).show()
            }
        }
    }

    override fun onDestroy() {
        super.onDestroy()
        try {
            btSocket?.close()
        } catch (e: IOException) {
            Log.e(TAG, "Error closing Bluetooth socket: ${e.message}")
```

```
      }
   }
}
```

Additional Files for the Arduino connections:

We cannot add the complete code so we are adding only the informative code for these sections.

Code Snippet 1 :

```
.class public final synthetic  MainActivity$$ExternalSyntheticLambda2;
.super Ljava / lang / Object;

# interfaces
  .implements Landroid / view / View$OnClickListener;

# instance fields
  .field public final synthetic f$0:   MainActivity;

# direct methods
  .method public synthetic constructor < init > (  MainActivity;) V
  .registers 2

invoke - direct {
  p0
}, Ljava / lang / Object; -> < init > () V

iput - object p1, p0,   MainActivity$$ExternalSyntheticLambda2;  f$0:   MainActivity;

return -void
  .end method

# virtual methods
  .method public final onClick(Landroid / view / View;) V
  .registers 3
```

```
iget - object v0, p0,MainActivity$$ExternalSyntheticLambda2; MainActivity;

invoke - static {
  v0,
  p1
},   MainActivity; -> $r8$lambda$5GWktFC4kHOqgPWoP8BIKBixuN4(  MainActivity;
Landroid / view / View;) V

return -void
  .end method
```

Code Snippet 2 :

```
.class public final synthetic MainActivity$$ExternalSyntheticLambda1;
.super Ljava / lang / Object;

# interfaces
        .implements Landroid / content / DialogInterface$OnClickListener;

# static fields
        .field public static final synthetic INSTANCE: MainActivity$$ExternalSyntheticLambda1;

# direct methods
        .method static synthetic constructor < clinit > () V
        .registers 1

new - instance v0,  MainActivity$$ExternalSyntheticLambda1;

invoke - direct {
        v0
}, MainActivity$$ExternalSyntheticLambda1; -> < init > () V

sput - object v0, MainActivity$$ExternalSyntheticLambda1; -> INSTANCE:
MainActivity$$ExternalSyntheticLambda1;
```

```
return -void
        .end method

        .method private synthetic constructor < init > () V
        .registers 1

invoke - direct {
        p0
}, Ljava / lang / Object; -> < init > () V

return -void
        .end method

# virtual methods
        .method public final onClick(Landroid / content / DialogInterface; I) V
        .registers 3

invoke - static {
        p1,
        p2
},  MainActivity; -> $r8$lambda$fA28pevyl4BDSGqsKIheyRIPrws(Landroid / content /
DialogInterface; I) V

return -void
        .end method
```

Code Snippet 3 :

```
.class public final   MainActivity;
.super Landroidx / appcompat / app / AppCompatActivity;
.source "MainActivity.kt"

# annotations
        .annotation runtime Lkotlin / Metadata;
d1 = {
```

"\u0000^\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\u0008\u0002\n\u0002\u0018\u0002\n\u0002\u0008\u0005\n\u0002\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u0002\u0008\u0005\n\u0002\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u0002\u0008\u0002\n\u0002\u0018\u0002\n\u0002\u0008\u0005\n\u0002\u0010\u0002\n\u0002\u0008\u0002\n\u0002\u0018\u0002\n\u0000\u0018\u00002\u00020\u0001B\u0005\u00a2\u0006\u0002\u0010\u0002J\u0008\u0010\u0017\u001a\u00020%H\u0002J\u0012\u0010&\u001a\u00020%2\u0008\u0010\'\u001a\u0004\u0018\u00010(H\u0014J\u0008\u0010$\u001a\u00020%H\u0002R\u001a\u0010\u0003\u001a\u00020\u0004X\u0086.\u00a2\u0006\u000e\n\u0000\u001a\u0004\u0008\u0005\u0010\u0006\"\u0004\u0008\u0007\u0010\u0008R\u000e\u0010\t\u001a\u00020\nX\u0082.\u00a2\u0006\u0002\n\u0000R\u0010\u0010\u000b\u001a\u0004\u0018\u00010\u000cX\u0082\u000e\u00a2\u0006\u0002\n\u0000R\u000e\u0010\r\u001a\u00020\u000eX\u0082.\u00a2\u0006\u0002\n\u0000R\u001a\u0010\u000f\u001a\u00020\u0010X\u0086.\u00a2\u0006\u000e\n\u0000\u001a\u0004\u0008\u0011\u0010\u0012\"\u0004\u0008\u0013\u0010\u0014R\u000e\u0010\u0015\u001a\u00020\u0016X\u0082.\u00a2\u0006\u0002\n\u0000R\u000e\u0010\u0017\u001a\u00020\u0018X\u0082.\u00a2\u0006\u0002\n\u0000R\u000e\u0010\u0019\u001a\u00020\u001aX\u0082\u0004\u00a2\u0006\u0002\n\u0000R\u000e\u0010\u001b\u001a\u00020\u001cX\u0082.\u00a2\u0006\u0002\n\u0000R\u000e\u0010\u001d\u001a\u00020\u001eX\u0082.\u00a2\u0006\u0002\n\u0000R\u001a\u0010\u001f\u001a\u00020 X\u0086.\u00a2\u0006\u000e\n\u0000\u001a\u0004\u0008
\u0010!\"\u0004\u0008\"\u0010#R\u000e\u0010$\u001a\u00020\u0018X\u0082.\u00a2\u0006\u0002\n\u0000\u00a8\u0006)"
}
d2 = {
" MainActivity;",
"Landroidx/appcompat/app/AppCompatActivity;",
"()V",
"biometricPrompt",
"Landroidx/biometric/BiometricPrompt;",
"getBiometricPrompt",
"()Landroidx/biometric/BiometricPrompt;",
"setBiometricPrompt",
"(Landroidx/biometric/BiometricPrompt;)V",

```
    "bluetoothAdapter",
    "Landroid/bluetooth/BluetoothAdapter;",
    "bluetoothSocket",
    "Landroid/bluetooth/BluetoothSocket;",
    "bt",
    "Landroid/bluetooth/BluetoothDevice;",
    "executor",
    "Ljava/util/concurrent/Executor;",
    "getExecutor",
    "()Ljava/util/concurrent/Executor;",
    "setExecutor",
    "(Ljava/util/concurrent/Executor;)V",
    "loading",
    " util;",
    "lock",
    "Lcom/google/android/material/button/MaterialButton;",
    "mUUID",
    "Ljava/util/UUID;",
    "mac",
    "Landroid/widget/TextView;",
    "mode",
    "promptInfo",
    "Landroidx/biometric/BiometricPrompt$PromptInfo;",
    "getPromptInfo",
    "()Landroidx/biometric/BiometricPrompt$PromptInfo;",
    "setPromptInfo",
    "(Landroidx/biometric/BiometricPrompt$PromptInfo;)V",
    "unlock",
    "",
    "onCreate",
    "savedInstanceState",
    "Landroid/os/Bundle;",
    "app_debug"
}
```

```
k = 0x1
mv = {
        0x1,
        0x5,
        0x1
}
xi = 0x30
        .end annotation

# instance fields
        .field public biometricPrompt: Landroidx / biometric / BiometricPrompt;

.field private bluetoothAdapter: Landroid / bluetooth / BluetoothAdapter;

.field private bluetoothSocket: Landroid / bluetooth / BluetoothSocket;

.field private bt: Landroid / bluetooth / BluetoothDevice;

.field public executor: Ljava / util / concurrent / Executor;

.field private loading:   util;

.field private lock: Lcom / google / android / material / button / MaterialButton;

.field private final mUUID: Ljava / util / UUID;

.field private mac: Landroid / widget / TextView;

.field private mode: Landroid / widget / TextView;

.field public promptInfo: Landroidx / biometric / BiometricPrompt$PromptInfo;

.field private unlock: Lcom / google / android / material / button / MaterialButton;
```

```
# direct methods
    .method public static synthetic $r8$lambda$5GWktFC4kHOqgPWoP8BIKBixuN4(
MainActivity; Landroid / view / View;) V
    .registers 2

invoke - static {
    p0,
    p1
},  MainActivity; -> onCreate$lambda - 1(  MainActivity; Landroid / view / View;) V

return -void
    .end method

    .method public static synthetic
$r8$lambda$8cGYcoaI3jYTdGh6ATEMnCfOQ7Y(Landroid / content / DialogInterface; I) V
    .registers 2

invoke - static {
    p0,
    p1
},  MainActivity; -> unlock$lambda - 2(Landroid / content / DialogInterface; I) V

return -void
    .end method

    .method public static synthetic $r8$lambda$fA28pevyl4BDSGqsKIheyRIPrws(Landroid /
content / DialogInterface; I) V
    .registers 2

invoke - static {
    p0,
    p1
},  MainActivity; -> lock$lambda - 3(Landroid / content / DialogInterface; I) V
```

```
return -void
        .end method

        .method public static synthetic $r8$lambda$rjdurLIyvGlauRSfdiVMkHerw3s(
MainActivity; Landroid / view / View;) V
        .registers 2

invoke - static {
        p0,
        p1
},   MainActivity; -> onCreate$lambda - 0(  MainActivity; Landroid / view / View;) V

return -void
        .end method

        .method public constructor < init > () V
        .registers 3

        .line 26
invoke - direct {
        p0
}, Landroidx / appcompat / app / AppCompatActivity; -> < init > () V

        .line 30
const -string v0, "00001101-0000-1000-8000-00805F9B34FB"

invoke - static {
        v0
}, Ljava / util / UUID; -> fromString(Ljava / lang / String;) Ljava / util / UUID;

move - result - object v0

const -string v1, "fromString(\"00001101-0000-1000-8000-00805F9B34FB\")"
```

```
invoke - static {
        v0,
        v1
}, Lkotlin / jvm / internal / Intrinsics; -> checkNotNullExpressionValue(Ljava / lang / Object;
Ljava / lang / String;) V

iput - object v0, p0,   MainActivity; -> mUUID: Ljava / util / UUID;

.line 26
return -void
        .end method

        .method public static final synthetic access$getLoading$p(  MainActivity;)   util;
.registers 2
        .param p0, "$this"
#   MainActivity;

.line 26
iget - object v0, p0,   MainActivity; -> loading:   util;

return -object v0
        .end method

        .method public static final synthetic access$unlock(  MainActivity;) V
        .registers 1
        .param p0, "$this"
#   MainActivity;

.line 26
invoke - direct {
        p0
},   MainActivity; -> unlock() V

return -void
```

```
        .end method

        .method private final lock() V
        .registers 7

        .line 127: cond_0
nop

        .line 128
const / 4 v0, 0x0

        : try_start_2
iget - object v1, p0,   MainActivity; -> bt: Landroid / bluetooth / BluetoothDevice;

if -eqz v1,: cond_3d

iget - object v2, p0,   MainActivity; -> mUUID: Ljava / util / UUID;

invoke - virtual {
        v1,
        v2
}, Landroid / bluetooth / BluetoothDevice; ->
createInsecureRfcommSocketToServiceRecord(Ljava / util / UUID;) Landroid / bluetooth /
BluetoothSocket;

move - result - object v1

iput - object v1, p0,   MainActivity; -> bluetoothSocket: Landroid / bluetooth / BluetoothSocket;

.line 129
if -nez v1,: cond_11

goto: goto_14
```

```
        : cond_11
invoke - virtual {
        v1
}, Landroid / bluetooth / BluetoothSocket; -> connect() V

        .line 130: goto_14
iget - object v1, p0,   MainActivity; -> bluetoothSocket: Landroid / bluetooth / BluetoothSocket;

if -nez v1,: cond_1a

move - object v1, v0

goto: goto_22

        : cond_1a
invoke - virtual {
        v1
}, Landroid / bluetooth / BluetoothSocket; -> isConnected() Z

move - result v1

invoke - static {
        v1
}, Ljava / lang / Boolean; -> valueOf(Z) Ljava / lang / Boolean;

move - result - object v1

        : goto_22
invoke - static {
        v1
}, Lkotlin / jvm / internal / Intrinsics; -> checkNotNull(Ljava / lang / Object;) V

invoke - virtual {
        v1
```

```
}, Ljava / lang / Boolean; -> booleanValue() Z

move - result v1

if -eqz v1,: cond_47

    .line 131
iget - object v1, p0,   MainActivity; -> mac: Landroid / widget / TextView;

if -eqz v1,: cond_37

const -string v2, "Connected"

check - cast v2, Ljava / lang / CharSequence;

invoke - virtual {
    v1,
    v2
}, Landroid / widget / TextView; -> setText(Ljava / lang / CharSequence;) V

    .line 132
goto: goto_5e

    .line 131: cond_37
const -string v1, "mac"
    v1
},   util; -> < init > (Landroid / app / Activity;) V

iput - object v0, p0,   MainActivity; -> loading:   util;

.line 47
invoke - static {}, Landroid / bluetooth / BluetoothAdapter; -> getDefaultAdapter() Landroid /
bluetooth / BluetoothAdapter;
```

```
move - result - object v0

const -string v1, "getDefaultAdapter()"

invoke - static {
        v0,
        v1
}, Lkotlin / jvm / internal / Intrinsics; -> checkNotNullExpressionValue(Ljava / lang / Object;
Ljava / lang / String;) V

iput - object v0, p0,   MainActivity; -> bluetoothAdapter: Landroid / bluetooth /
BluetoothAdapter;

.line 48
const / 4 v1, 0x0

if -eqz v0,: cond_c4

const -string v2, "00:21:09:00:1B:AA"

invoke - virtual {
        v0,
        v2
}, Landroid / bluetooth / BluetoothAdapter; -> getRemoteDevice(Ljava / lang / String;) Landroid
/ bluetooth / BluetoothDevice;

move - result - object v0

const -string v2, "bluetoothAdapter.getRemoteDevice(\"00:21:09:00:1B:AA\")"

invoke - static {
        v0,
        v2
}, Lkotlin / jvm / internal / Intrinsics; -> checkNotNullExpressionValue(Ljava / lang / Object;
```

```
Ljava / lang / String;) V

iput - object v0, p0,   MainActivity; -> bt: Landroid / bluetooth / BluetoothDevice;

.line 49
iput - object v1, p0,   MainActivity; -> bluetoothSocket: Landroid / bluetooth / BluetoothSocket;

.line 50
const / 4 v0, 0x0

    .line 51
    .local v0, "counter": I
move - object v2, p0

check - cast v2, Landroid / content / Context;

invoke - static {
    v2
}, Landroidx / core / content / ContextCompat; -> getMainExecutor(Landroid / content /
Context;) Ljava / util / concurrent / Executor;

move - result - object v2

const -string v3, "getMainExecutor(this)"
check - cast v5, Landroidx / biometric / BiometricPrompt$AuthenticationCallback;

.line 52
invoke - direct {
    v2,
    v3,
    v4,
    v5
}, Landroidx / biometric / BiometricPrompt; -> < init > (Landroidx / fragment / app /
FragmentActivity; Ljava / util / concurrent / Executor; Landroidx / biometric /
```

```
BiometricPrompt$AuthenticationCallback;) V

invoke - virtual {
        p0,
        v2
},   MainActivity; -> setBiometricPrompt(Landroidx / biometric / BiometricPrompt;) V

        .line 70
new - instance v2, Landroidx / biometric / BiometricPrompt$PromptInfo$Builder;

invoke - direct {
        v2
}, Landroidx / biometric / BiometricPrompt$PromptInfo$Builder; -> < init > () V

        .line 71
const -string v3, "Authentication"

check - cast v3, Ljava / lang / CharSequence;

invoke - virtual {
        v2,
        v3
}, Landroidx / biometric / BiometricPrompt$PromptInfo$Builder; -> setTitle(Ljava / lang /
CharSequence;) Landroidx / biometric / BiometricPrompt$PromptInfo$Builder;

move - result - object v2

        .line 72
const -string v3, "Please scan your fingerprint in order to Unlock the door"

check - cast v3, Ljava / lang / CharSequence;

invoke - virtual {
        v2,
```

```
    v3
}, Landroidx / biometric / BiometricPrompt$PromptInfo$Builder; -> setSubtitle(Ljava / lang /
CharSequence;) Landroidx / biometric / BiometricPrompt$PromptInfo$Builder;

move - result - object v2

    .line 73
const -string v3, "Cancel"

check - cast v3, Ljava / lang / CharSequence;

invoke - virtual {
    v2,
    v3
}, Landroidx / biometric / BiometricPrompt$PromptInfo$Builder; ->
setNegativeButtonText(Ljava / lang / CharSequence;) Landroidx / biometric /
BiometricPrompt$PromptInfo$Builder;

move - result - object v2

    .line 74
invoke - virtual {
    v2
}, Landroidx / biometric / BiometricPrompt$PromptInfo$Builder; -> build() Landroidx /
biometric / BiometricPrompt$PromptInfo;

move - result - object v2

const -string v3, "Builder()\n          .setTitle(\"Authentication\")\n          .setSubtitle(\"Please
scan your fingerprint in order to Unlock the door\")\n
.setNegativeButtonText(\"Cancel\")\n          .build()"

invoke - static {
    v2,
```

```
        v3
}, Lkotlin / jvm / internal / Intrinsics; -> checkNotNullExpressionValue(Ljava / lang / Object;
Ljava / lang / String;) V

        .line 70
invoke - virtual {
        p0,
        v2
},   MainActivity; -> setPromptInfo(Landroidx / biometric / BiometricPrompt$PromptInfo;) V

        .line 75
iget - object v2, p0,   MainActivity; -> unlock: Lcom / google / android / material / button /
MaterialButton;

if -eqz v2,: cond_be

new - instance v3,   MainActivity$$ExternalSyntheticLambda3;

invoke - direct {
        v3,
        p0
},   MainActivity$$ExternalSyntheticLambda3; -> < init > (  MainActivity;) V

invoke - virtual {
        v2,
        v3
}, Lcom / google / android / material / button / MaterialButton; -> setOnClickListener(Landroid /
view / View$OnClickListener;) V

        .line 80
iget - object v2, p0,   MainActivity; -> lock: Lcom / google / android / material / button /
MaterialButton;

if -eqz v2,: cond_b8
```

```
new - instance v1,   MainActivity$$ExternalSyntheticLambda2;

invoke - direct {
        v1,
        p0
},   MainActivity$$ExternalSyntheticLambda2; -> < init > (  MainActivity;) V

invoke - virtual {
        v2,
        v1
}, Lcom / google / android / material / button / MaterialButton; -> setOnClickListener(Landroid /
view / View$OnClickListener;) V

        .line 86
return -void

        .line 80: cond_b8
const -string v2, "lock"

invoke - static {
        v2
}, Lkotlin / jvm / internal / Intrinsics; -> throwUninitializedPropertyAccessException(Ljava /
lang / String;) V

throw v1

        .line 75: cond_be
const -string v2, "unlock"

invoke - static {
        v2
}, Lkotlin / jvm / internal / Intrinsics; -> throwUninitializedPropertyAccessException(Ljava /
lang / String;) V
```

```
throw v1

    .line 48
    .end local v0 # "counter": I: cond_c4
const -string v0, "bluetoothAdapter"

invoke - static {
    v0
}, Lkotlin / jvm / internal / Intrinsics; -> throwUninitializedPropertyAccessException(Ljava /
lang / String;) V

throw v1
    .end method

    .method public final setBiometricPrompt(Landroidx / biometric / BiometricPrompt;) V
    .registers 3
    .param p1, "<set-?>"
# Landroidx / biometric / BiometricPrompt;

const -string v0, "<set-?>"

invoke - static {
    p1,
    v0
}, Lkotlin / jvm / internal / Intrinsics; -> checkNotNullParameter(Ljava / lang / Object; Ljava /
lang / String;) V

    .line 28
iput - object p1, p0,   MainActivity; -> biometricPrompt: Landroidx / biometric /
BiometricPrompt;

return -void
    .end method
```

```
    .method public final setExecutor(Ljava / util / concurrent / Executor;) V
    .registers 3
    .param p1, "<set-?>"
# Ljava / util / concurrent / Executor;

const -string v0, "<set-?>"

invoke - static {
    p1,
    v0
}, Lkotlin / jvm / internal / Intrinsics; -> checkNotNullParameter(Ljava / lang / Object; Ljava /
lang / String;) V

    .line 27
iput - object p1, p0,   MainActivity; -> executor: Ljava / util / concurrent / Executor;

return -void
    .end method

    .method public final setPromptInfo(Landroidx / biometric / BiometricPrompt$PromptInfo;)
V
    .registers 3
    .param p1, "<set-?>"
# Landroidx / biometric / BiometricPrompt$PromptInfo;

const -string v0, "<set-?>"

invoke - static {
    p1,
    v0
}, Lkotlin / jvm / internal / Intrinsics; -> checkNotNullParameter(Ljava / lang / Object; Ljava /
lang / String;) V
```

```
        .line 29
iput - object p1, p0,   MainActivity; -> promptInfo: Landroidx / biometric /
BiometricPrompt$PromptInfo;

return -void
        .end method
.class public final LnoKey/MainActivity;
.super Landroidx/appcompat/app/AppCompatActivity;
.source "MainActivity.kt"



# annotations
.annotation runtime Lkotlin/Metadata;
        d1 = {

"\u0000^\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\u0008\u0002\n\u0002\u0018\u0
002\n\u0002\u0008\u0005\n\u0002\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u0000\n\u000
2\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u0002\u0008\u0005\n\u0002\u0018\u0002\n\u0
000\n\u0002\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u0000\n\u0002\u0018\u0002\n\u000
2\u0008\u0002\n\u0002\u0018\u0002\n\u0002\u0008\u0005\n\u0002\u0010\u0002\n\u0002\u00
08\u0002\n\u0002\u0018\u0002\n\u0000\u0018\u00002\u00020\u0001B\u0005\u00a2\u0006\u0
002\u0010\u0002J\u0008\u0010\u0017\u001a\u00020%H\u0002J\u0012\u0010&\u001a\u00020
%2\u0008\u0010\'\u001a\u0004\u0018\u00010(H\u0014J\u0008\u0010$\u001a\u00020%H\u000
2R\u001a\u0010\u0003\u001a\u00020\u0004X\u0086.\u00a2\u0006\u000e\n\u0000\u001a\u000
4\u0008\u0005\u0010\u0006\"\u0004\u0008\u0007\u0010\u0008R\u000e\u0010\t\u001a\u00020
\nX\u0082.\u00a2\u0006\u0002\n\u0000R\u0010\u0010\u000b\u001a\u0004\u0018\u00010\u00
0cX\u0082\u000e\u00a2\u0006\u0002\n\u0000R\u000e\u0010\r\u001a\u00020\u000eX\u0082.\u
00a2\u0006\u0002\n\u0000R\u001a\u0010\u000f\u001a\u00020\u0010X\u0086.\u00a2\u0006\u
000e\n\u0000\u001a\u0004\u0008\u0011\u0010\u0012\"\u0004\u0008\u0013\u0010\u0014R\u0
00e\u0010\u0015\u001a\u00020\u0016X\u0082.\u00a2\u0006\u0002\n\u0000R\u000e\u0010\u0
017\u001a\u00020\u0018X\u0082.\u00a2\u0006\u0002\n\u0000R\u000e\u0010\u0019\u001a\u0
0020\u001aX\u0082\u0004\u00a2\u0006\u0002\n\u0000R\u000e\u0010\u001b\u001a\u00020\u0
01cX\u0082.\u00a2\u0006\u0002\n\u0000R\u000e\u0010\u001d\u001a\u00020\u001cX\u0082.\
u00a2\u0006\u0002\n\u0000R\u001a\u0010\u001e\u001a\u00020\u001fX\u0086.\u00a2\u0006\
```

u000e\n\u0000\u001a\u0004\u0008
\u0010!\"\u0004\u0008\"\u0010#R\u000e\u0010$\u001a\u00020\u0018X\u0082.\u00a2\u0006\u0002\n\u0000\u00a8\u0006)"

```
    }
    d2 = {
        "LnoKey/MainActivity;",
        "Landroidx/appcompat/app/AppCompatActivity;",
        "()V",
        "biometricPrompt",
        "Landroidx/biometric/BiometricPrompt;",
        "getBiometricPrompt",
        "()Landroidx/biometric/BiometricPrompt;",
        "setBiometricPrompt",
        "(Landroidx/biometric/BiometricPrompt;)V",
        "bluetoothAdapter",
        "Landroid/bluetooth/BluetoothAdapter;",
        "bluetoothSocket",
        "Landroid/bluetooth/BluetoothSocket;",
        "bt",
        "Landroid/bluetooth/BluetoothDevice;",
        "executor",
        "Ljava/util/concurrent/Executor;",
        "getExecutor",
        "()Ljava/util/concurrent/Executor;",
        "setExecutor",
        "(Ljava/util/concurrent/Executor;)V",
        "loading",
        "LnoKey/util;",
        "lock",
        "Lcom/google/android/material/button/MaterialButton;",
        "mUUID",
        "Ljava/util/UUID;",
        "mac",
        "Landroid/widget/TextView;",
```

```
            "mode",
            "promptInfo",
            "Landroidx/biometric/BiometricPrompt$PromptInfo;",
            "getPromptInfo",
            "()Landroidx/biometric/BiometricPrompt$PromptInfo;",
            "setPromptInfo",
            "(Landroidx/biometric/BiometricPrompt$PromptInfo;)V",
            "unlock",
            "",
            "onCreate",
            "savedInstanceState",
            "Landroid/os/Bundle;",
            "app_debug"
        }
        k = 0x1
        mv = {
            0x1,
            0x5,
            0x1
        }
        xi = 0x30
.end annotation


# instance fields
.field public biometricPrompt:Landroidx/biometric/BiometricPrompt;

.field private bluetoothAdapter:Landroid/bluetooth/BluetoothAdapter;

.field private bluetoothSocket:Landroid/bluetooth/BluetoothSocket;

.field private bt:Landroid/bluetooth/BluetoothDevice;

.field public executor:Ljava/util/concurrent/Executor;
```

Description of the Code:

The code we implemented is a simple Android application that interacts with an Arduino board over Bluetooth to control a solenoid lock. It has a main activity that contains two buttons, "Lock" and "Unlock", and uses the Bluetooth Adapter and Bluetooth Socket classes to establish a connection with the Arduino board.

When the user clicks on the "Lock" button, the app sends the letter "L" over Bluetooth to the Arduino, which triggers the solenoid lock to lock. When the user clicks on the "Unlock" button, the app sends the letter "U" over Bluetooth to the Arduino, which triggers the solenoid lock to unlock.

To implement this functionality, we first checked if the device supports Bluetooth by calling the BluetoothAdapter.getDefaultAdapter() method. If this returns null, it means the device doesn't support Bluetooth, and we display an error message to the user.

Next, we checked if Bluetooth is enabled on the device by calling the isEnabled() method on the Bluetooth Adapter object. If Bluetooth is not enabled, we request the user to enable it by displaying a message and starting an intent to open the Bluetooth settings activity.

If Bluetooth is enabled, we attempt to establish a connection with the Arduino board by calling the connect() method on a Bluetooth Socket object. This method takes the MAC address of the Bluetooth device we want to connect to as a parameter. In our case, we hard-coded the MAC address of the Arduino board.

Once the connection is established, we can send data to the Arduino board using the write() method on the Output Stream object of the Bluetooth Socket. We also receive data from the Arduino board using the read() method on the Input Stream object.
Finally, we added onClickListeners to the "Lock" and "Unlock" buttons to send the corresponding letters over Bluetooth to the Arduino board when clicked.

This implementation is just a basic example of how to interact with an Arduino board over Bluetooth using an Android application. In a real-world application, we would need to add error handling, input validation, and more robust Bluetooth connectivity handling.

Hardware specifications:

Solenoid lock:

Fig 4.1 Solenoid Lock

It is to be had in opening inside the force on mode type and bolting and protecting inside the strength-on mode type, which might be utilized specifically for conditions. The strength-on opening kind permits opening handiest while the solenoid is fuelled on. An entryway with this sort is secured and not opened instance of energy disappointment or twine detachment, guaranteeing incredible assurance. This sort is utilized particularly for places requiring wrongdoing avoidance. The strength-on locking kind can bolt an entryway while the solenoid is fuelled on. In the event that the strength is detached, the entryway is opened. This sort opens the entryway in the event of string separation because of a fire or spot of destiny, and it is utilized for crisis exits through which putting out fires interest or departure need to specially be made rather than assurance for wrongdoing counteraction. The keeping type performs two tasks, bolting and opening with the guide of applying a beneficial or awful heartbeat voltage to the solenoid, and keeps up the no-power country in every job. This sort capacities energy saving because of the reality it's far trivial to consistently power the solenoid on. For the ceaseless score and the discontinuous score, the relentless score is planned to take care of an appraised voltage power continually for quite a long time without surpassing a focused-on temperature

upward push limit, and the irregular rating is planned as an approach to take care of an itemized voltage most straightforward for a predetermined time length without surpassing a definite temperature upward push confine.

Specification of solenoid lock:

➢ Operating voltage :12vdc

➢ Draws 650 mA at 12 volts 500 mA at 9 volts

➢ Designed for 1-10 seconds for long activation time

➢ Wirelength 222.5 mA

Features:

- Iron Body Material.
- High excellent extremely compact electric powered lock.
- Rustproof, durable, safe, handy to use.
- Suction which tightly sucks the iron, for this reason locking the door.
- Applicable for being installed inside the breakout door or hearth door digital controlled gadget.
- This Lock adopts the principle of electric magnetism.
- Operating Voltage: 12V.
- Rated Current: zero.8 A.
- Holding Force: zero.25 kg.
- Weight: a hundred and fifty gm.
- Cable Length: 35 cm (with connector).

Arduino Nano:

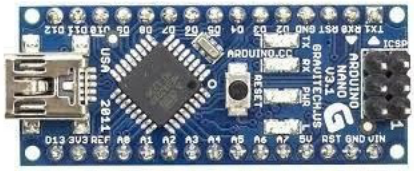


Fig 4.2 Arduino Nano Circuit Board

Arduino nano is an open-source gadgets prototyping stage dependent on bendy, clean- to utilize equipment and programming. It's alleged for craftsmen, architects, specialists, and totally everybody entranced in establishing intelligent articles or conditions. It's an open-supply substantial figuring stage dependent on a micro-controller board, and an advancement climate for composing programming program for the board. In basic words, Arduino is a little micro-controller board with a USB fitting to associate with your

pc and some of association attachments that might be wired up to outside gadgets, for example, engines, transfers, light sensors, laser diodes, amplifiers, mouthpieces, etc., They can both be fuelled through the USB association from the PC or from a 9V battery. They can be controlled from the PC or modified through the PC and afterward separated and permitted to work freely electronic devices are getting reduced, bendy and modest which may have the capacity to do more prominent trademark when contrasted with their archetypes that came upon to cowl more region, turned out expensive with the ability to perform less highlights. Specialists consistently endeavour to present advancement in computerization that requires least endeavour and offers most yield. The micro-controller was conveyed in the gadgets business with the explanation of making our commitments clean that accompany even a faraway association with computerization in any capacity. Microcontrollers are broadly used in installed frameworks and make gadgets work with regards to our necessities and prerequisite. Arduino Uno is a completely prized expansion inside the gadgets that comprises of USB interface, 14 virtual I/O pins, 6 simple pins, and Atmega328 micro-controller.

Arduino is an open-source electronics prototyping platform based on flexible, easy-to- use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino received an Honory Mention in the Digital Communities section of the 2006 Ars Electronica Prix. Credits Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a smallest, complete, and breadboard friendly. It has everything that Diecimila/Duemilanove has (electrically) with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano is automatically sense and switch to the higher potential source of power, there is no need for the power select jumper. Nano's got the breadboard-ability of the Boarduino and the Mini+USB with smaller footprint than either, so users have more breadboard space. It's got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, GND on one top, power and ground on the other). This new version 3.0 comes with ATMEGA328 which offer more programming and data memory space. It is two layers. That make it easier to hack and more affordable.

Relay module:

Fig 4.3 Relay Lock Module

A Relay is an electromechanical machine that can be used to make or damage an electrical connection. It consists of a bendy transferring mechanical phase which can be manged

electronically via an electromagnet, basically, a relay is simple like a mechanical change however you can manipulate it with a digital sign rather of mutually turning it on or off. Again, this working precept of relay suits solely for the electromechanical relay.

There are many sorts of relay and every relay has its personal application, a standard, ana commonly used relay is made up of electromagnets which in normally used as a switch. Dictionary says that relay ability the act of passing something from one component to another, the identical which means can be utilized to this gadget due to the fact the sign acquired from one facet of the gadget controls the switching operation on the different side. So, relay is a change which controls (open and close) circuits electromechanically. The principal operation of this machine is to make or damage contact with the assist of assign except any human involvement in order to change it ON or OFF. It is in the main used to manipulate an excessive powered circuit the usage of a low electricity signal. Generally, a DC sign is used to manage the circuit which is pushed by means of excessive voltage like controlling AC domestic home equipment with DC indicators from microcontrollers.

A relay basically allows a relatively low voltage to easily control higher power circuits. A relay accomplishes this by using the 5V outputted from an Arduino pin to energize the electromagnet which in turn closes an internal, physical switch to turn on or off a higher power circuit. relay is an electrically operated switch. ... Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. A 5v relay is an automatic switch that is commonly used in an automatic control circuit and to control a high-current using a low- current signal. The input voltage of the relay signal ranges from 0 to 5V.

Types of Relays:

- Electromagnetic Relays.
- Latching Relays.
- Electronic Relays.
- Non-Latching Relays.
- Reed Relays.
- High-Voltage Relays.

- Small Signal Relays.
- Time Delay Relays.

Bluetooth HC 05 Module:

Fig 4.4 Bluetooth Chip Module with Connector Pins

It is used for many purposes like wifi headset, wifi controllers, wifi mouse, wifi keyboard and many application. It has varied up to <100m which relies upon upon transmitter and receiver, atmosphere, geographic & city conditions. It is IEEE 802.15.1 standardized protocol, via which one can construct wi-fi Personal Area Network (PAN), It makes use of frequency-hopping unfold spectrum (FHSS) radio technological know-how to ship records over air, It makes use of serial verbal exchange to talk with devices. It communicates with microcontroller the use of serial port (USART). HC-05 is a Bluetooth module which is designed for wi-fi communication. This module can be used in a grasp or slave configuration.

HC-05 has crimson LED which shows connection status, whether the Bluetooth is related or not. Before connecting to HC-05 module this crimson LED blinks always in a periodic manner. When it receives linked to any different Bluetooth device, its blinking slows down to two seconds.

This module works on 3.3 V. We can join 5V provide voltage as nicely due to the fact the module has on board 5 to 3.3 V regulator. As HC-05 Bluetooth module has V stage for RX/TX and microcontroller can become aware of 3.3 V level, so, no need to shift transmit degree of HC-05 module. But we want to shift the transmit voltage

stage from microcontroller to RX of HC-05 module. After pairing two Bluetooth devices, open terminal software program (e.g., Tera term, Realter etc.) in PC, and pick out the port the place we have related USB to serial module. Also pick default baud charge of 9600 BPS. In clever phone, open Bluetooth terminal utility and join to paired machine HC-05. It is easy to communicate, we simply have to kind in the Bluetooth terminal utility of smartphone. Characters will get despatched wirelessly to Bluetooth module HC-OS. HC-05 will robotically transmit it serially to the PC, which will show up on terminal. Same way we can ship statistics from PC to smart phone. It is easy to communicate, we simply have to kind in the Bluetooth terminal utility of smartphone. Characters Will get to Bluetooth HC-OS. Will robotically transmit it serially to the PC, which will show up on terminal. way we can ship statistics from PC to smartphone. Bluetooth is a technological know-how for wi-fi communicate It is designed to substitute cable connections. It makes use of serial verbal exchange to talk with devices. It communicates with microcontroller the use of serial port (USART). Usually, it connects small units ike cellular phones, PDAs and TVs the usage of a short range wi-fi connection to alternate documents. It makes use of the 2 ,4SGHz frequency band. The connection can be point-to-point or multi-point the place the most vary is 10 meters. The switch charge of the statistics is 1 Mbps.

Connecting wires:

Fig 4.5 Connecting Wires

Present day non-steel sheathed links comprising of (US and Canadian) types NMB and NMC. Incorporate to 4 wires covered with thermoplastic protection, in addition to a bare wire for establishing (holding), encompassed with the guide of an adaptable plastic coat. A few forms envelop the man or lady conductors by paper before the plastic coat is done.

Unique adaptations of non-metal sheathed links, along with US Type UF, are intended for direct underground entombment (consistently with isolated mechanical security) or outside use where exposure to bright radiation (UV) is a chance. These links change in having a dampness safe creation, missing paper or distinctive retentive fillers, and being figured for UV opposition.

Elastic like fake polymer protection is used in modern links and energy links. Set up underground because of its boss dampness obstruction.

Protected links are appraised via their suitable working voltage and their most extreme working temperature at the conductor surface. A link may likewise convey more than one use rankings for programs, as an occurrence, one rating for dry establishments and some other while presented to dampness or oil By and large, single conduit developing line in little sizes is strong rope, in light of the fact that the wiring isn't constantly needed to be truly adaptable. Building twine conductors huge than 10 AWG (or around 6 mm^2) are abandoned for adaptability sooner or later of establishment, however, aren't adequately malleable to use as hardware twine.

Links for mechanical, modern and flats may moreover contain many protected channels in a standard coat, with helical tape metallic or aluminium shield, or metallic twine reinforcement, and perhaps too a customary PVC or lead coat for assurance from dampness and substantial damage. Links implied for very bendy supplier or in marine projects can be covered by means of woven bronze wires. Force or correspondence 17 links (e.g., PC organizing) that are steered in or through air overseeing spaces (plenums) of work environment homes is needed under the model construction law to be either encased in metal course or evaluated for low fire and smoke fabricating.

A couple of mechanical uses in metallic generators and similar warm conditions, no characteristic texture offers pleasant transporter. Links protected with packed mica pieces are incidentally utilized. Another state of exorbitant temperature link is a mineral protected link, with man or lady channels situated inside a copper tube and the distance brimming with magnesium oxide powder. The total gathering is drawn right down to more modest sizes, in this way compacting the powder. Such links have an authorized chimney opposition score and are more prominent profoundly valued than non-chimney evaluated link. They have little adaptability and carry on more noteworthy like firm channel instead of bendy links.

The environmental factors of the mounted wires decide how bounty current a link is approved to convey. Since different conductors packaged in a link can't use heat as easily as single protected conductors, the ones circuits are constantly evaluated at a lessening "ampacity".

Tables in electrical security codes give the greatest suitable forefront fundamentally dependent on size of channel, voltage capacity, protection type and thickness, and the temperature rating of the actual link. The admissible bleeding edge may likewise be unique for wet or dry spots, for good (storage room) or cool (underground) areas. In a run of link through a few zones, the part with the most minimal score turns into the rating of the general run.

Links commonly are made sure about with exceptional fittings wherein they enter electric gear; this might be a simple screw cinch for jacketed links in a dry spot, or a polymer-gasketed link connector that consequently draws in the protective layer of a defensively covered link and presents a waterproof association. Extraordinary link fittings can be done to keep touchy gases from streaming in the inside of jacketed links,

in which the link passes through zones wherein combustible gases are blessing. To forestall relaxing of the associations of character conductors of a link, links should be upheld near their front to contraptions and at ordinary stretches close by their runs.

Battery:

Fig 4.6 - Lithium Ion Battery 5V

A Battery Is a Tools made Out of One or Extra Electrochemical Cells with Outside Associations And Are Furnished To Power Electrical Gadgets Such As Flash Lighting apparatuses, Mobiles, And Electric Vehicles. At the point when The Battery Is Providing Electric controlled Energy Its Beneficial Terminal Is The Cathode And Its Bad Terminal Is The Anode The Terminal Stamped Poor Is The Supply Of The Electrons That Will Drift Via An External Electric fuelled Circuit To The Tremendous Terminal When A Battery Is Related To Outside Electric Load, A Redox Reaction Converts he High energy Reactants To The Low energy Merchandise, And The Free Power Distinction Is Brought To External Circuit Known as Electric Electricity Verifiably The Term "battery" Particularly Referred To Device made Out of More than one Cells, But The Usage Is Developed To Encompass Devices Composed Of An Unmarried Cellular . These batteries are partitioned into: Fundamental: (unmarried-use or "disposable") batteries are applied once and masterminded; the terminal materials are in an indirect way adjusted all through delivery. Typical occurrences of imperative batteries are solvent battery applied for spotlights and an enormous assortment of reduced electronic contraptions. Auxiliary BATTERIES: might be cleared and might be resuscitated on novel occasions using a reasonable applied electric controlled buoy; the essential venture of anodes might be re- set up with the guide of switch accept the way things are. Models contains the lead- 15 destructive batteries applied in vehicles and lithium batteries applied for devices, for example, PCs and cell phones. Batteries are in various sizes and styles, from little cells used to control listening gadgets and wristwatches to little, thin and gentle cells applied in cutting

edge cell phones, to huge lead destructive batteries or lithium-molecule batteries in the vehicles, and at a biggest incredible, wonderful battery bank the elements of rooms that supply save or debacle potential to cell phone trade and PC worker ranches.

As indicated by using the year 2005 gauge, the general battery industry creates round US$548 billion in deals every year, with 6% yearly improvement. Batteries have lower specific (vitality according to unit mass) than ordinary powers, for instance, gasoline.

Software specifications:

Arduino software:

Fig 4.7 Arduino Software for Windows version 2.1

Arduino is an open-source gadgets stage upheld simple to-utilize equipment and programming.

Arduino sheets can understand inputs - light on a sensor, a finger on a catch, or a Twitter message and switch it into a yield - actuating an engine, turning on a LED, distributing something on the Web. you'll mention to your board what to attempt to by sending a gaggle of directions to the microcontroller on the board. to embrace to so you utilize the Arduino programming language (considering Irking), and consequently the Arduino Software (IDE), upheld Processing.

Throughout the long-term Arduino has been the mind of thousands of tasks, from ordinary items to complex logical instruments. An overall network of creators - understudies, specialists, craftsmen, software engineers, and experts - has accumulated around this open-source stage, their commitments have amounted to an unfathomable measure of available information which might be of extraordinary assistance to learners and specialists the same.

Arduino was conceived at the Ivrea Interaction Design Institute as a simple device for quick prototyping, pointed toward understudies without a foundation in hardware and programming. As soon in light of the fact that it arrived at a more extensive network, the Arduino board began changing to adjust new requirements difficulties, separating.

CHAPTER 5

System Design and Analysis

This chapter provides a set of diagrams to help understand the project structure and workflow

in a better manner.

Architecture Design:
5.1 UML Class Diagrams

A Class Diagram is a type of UML diagram that represents the static structure of a system in terms of its classes, attributes, methods, and relationships. It is a visual representation of the objects in a system and how they are related to each other.

In the context of an Arduino-based smart door lock system, the Class Diagram can help to represent the objects and their relationships in the mobile application. The high-level Class Diagram of the system shows the most important objects in the mobile app, which are User, Friend, Smart Lock, and Activity.

The User class represents a user of the mobile app who has the ability to interact with the smart door lock system. This class contains attributes such as name, email, and password, as well as methods for adding and removing friends, and managing their own profile.

The Friend class represents a user's friends who are authorized to access the smart door lock system. This class contains attributes such as name and email, as well as methods for managing friend requests and permissions.

The Smart Lock class represents the physical door lock that is connected to the Arduino board. This class contains attributes such as status (locked or unlocked), access control settings, and methods for controlling the lock, such as lock, unlock, and check status.

The Activity class represents the user's activities within the mobile app, such as logging in, changing settings, and accessing the door lock. This class contains attributes such as date and time, and methods for recording and retrieving activities.

The relationships among these objects are also represented in the Class Diagram. For example, each User can add many Friends, which is represented by the association between the User and Friend classes. Each Smart Lock can have many Activities associated with it, which is represented by the association between the Smart Lock and Activity classes.

In summary, the Class Diagram is a useful tool for visualizing the objects and their relationships in the mobile application of an Arduino-based smart door lock system. It helps to clearly show the classes of the system, their interrelationships, and the operations and attributes of the classes, which can aid in the design and development of the system.

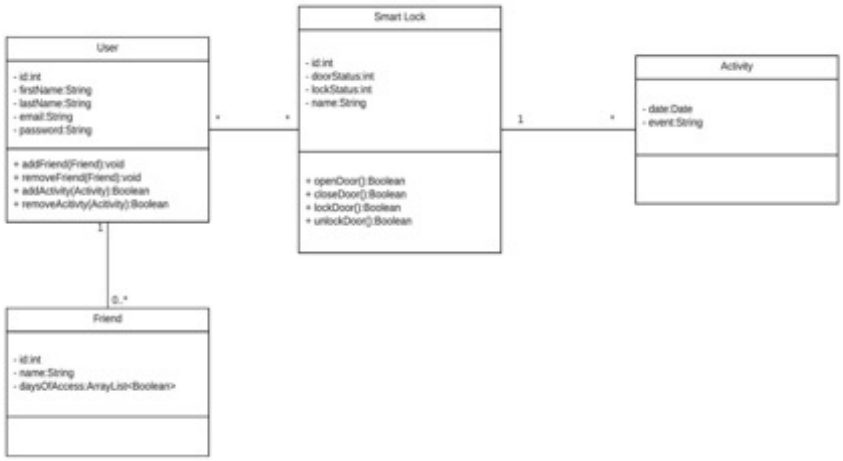


Figure 5.1 Class Diagram for Locking Mechanism

5.2 Components Diagrams

The Components Diagram in the context of an Arduino-based smart door lock system represents the interrelations between the various components involved in the functioning of the system.

The three main components in this system are:

Door Lock - This is the on-board logical unit mounted on the door. It is responsible for receiving commands from the cloud and processing them to lock or unlock the door. The Door Lock component could consist of various sub-components, such as a microcontroller, a servo motor, a sensor for detecting the presence of an authorized user, and an LED or sound module for indicating the status of the lock.

Mobile App - This is an application that runs on a mobile device connected to the internet. The Mobile App component is responsible for providing a user interface for interacting with the system, such as sending commands to unlock or lock the door, viewing the status of the lock, and registering authorized users. The Mobile App component could be built using various technologies, such as native mobile app development or cross-platform frameworks.

Cloud - This is an operating system running our software, which is responsible for authenticating faces and sending commands from the Mobile App to the on-board unit. The Cloud component could be hosted on a remote server and could consist of various sub-components, such as a facial recognition engine, a database for storing user profiles, and an API for communicating with the Mobile App and the Door Lock components.

The Components Diagram shows how these three components are interconnected and how they interact with each other. For example, the Mobile App sends commands to the Cloud, which authenticates the user's face and sends commands to the Door Lock to unlock or lock the door. The Door Lock component also sends status updates to the Mobile App, indicating whether the door is currently locked or unlocked.

In summary, the Components Diagram provides a high-level view of the architecture of the smart door lock system, illustrating the relationship between the Door Lock, Mobile App, and Cloud components. It allows the team to better understand how the different parts of the system interact with each other, helping them to identify potential issues and opportunities for improvement.

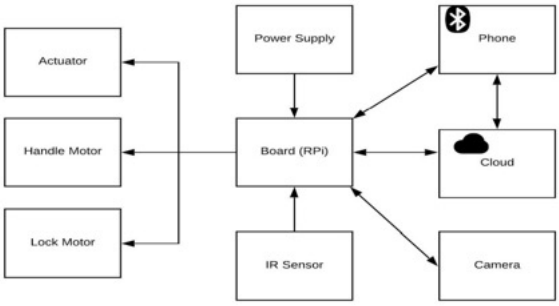
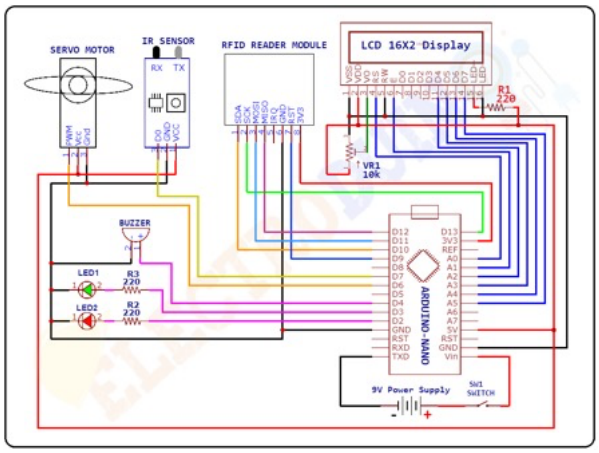


Figure 5.2 Components Diagram

5.3 State Machine Diagram

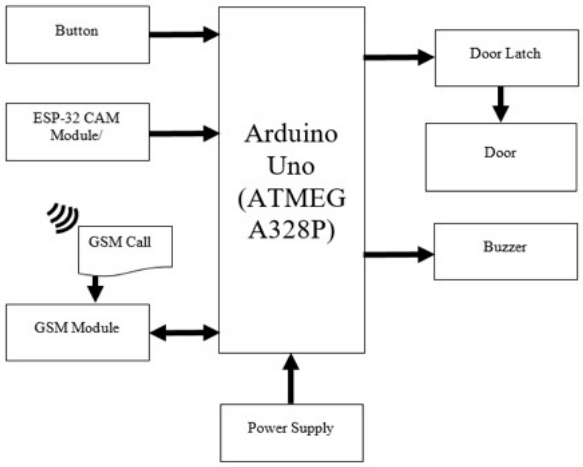
State Diagrams are particularly useful for complex components who can't fully be represented in a Class Diagram. UML state machine diagrams depict the various states that an object may be in and the transitions between those states [18]. For our system, we are going to build a state machine diagram that indicates the different states of our on-board logical unit, which will run an algorithm that controls the state of the door based on commands from the mobile app or presence of a person next to the door. An initial ideation of a state machine diagram is shown in Figure 6.3



5.3 State Machine Diagram for the Smart Lock system mounted on door.

Data Flow Diagram:

5.4 Flow Chart



5.4 Data Flow Architecture for Smart Lock System

In this flowchart, the system starts by reading input from a sensor, which could be a proximity sensor, fingerprint scanner, or some other type of authentication mechanism. The system then checks to see if an authorized user is present, and if so, it activates a servo motor to unlock the door. If the door doesn't open, the system alerts the user with a sound or light.

Data flow in an Arduino-based smart door unlock system refers to the movement of data between the various components of the system. For example, the sensor data is read by the microcontroller, which then processes the data and sends signals to the servo motor to unlock the door. Here is a possible data flow diagram for the same smart door unlock system:

Sensor input -> Microcontroller -> Servo motor -> Door unlock/lock Mechanism.

In this diagram, the sensor input is received by the microcontroller, which processes the data and sends signals to the servo motor to unlock or lock the door as required. The door unlock/lock mechanism is the destination of the data flow, as it is where the physical action of unlocking or locking the door takes place.

CHAPTER 6

EXPERIMENTAL RESULTS

A smart lock using IoT and Android would likely involve a physical lock that can be controlled remotely via an internet connection. The lock might have a built-in Wi-Fi or Bluetooth connection, or it might connect to a separate IoT device that controls the lock and communicates with an Android app.

6.1 Working of the IOT based Smart Lock

- The Android app would allow the user to remotely lock and unlock the door, as well as monitor the status of the lock (e.g., whether it is currently locked or unlocked). The app might also allow the user to set up access codes or other authentication methods for unlocking the door, such as fingerprint recognition or facial recognition.

- In terms of the implementation details, a smart lock using IoT and Android would likely involve programming the lock and the IoT device to communicate with each other over a wireless network. This could involve using a protocol like MQTT or HTTP to send commands and data between the devices.

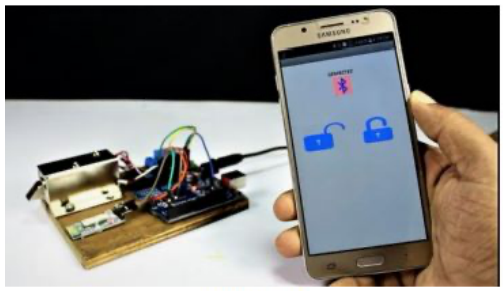


Figure 2

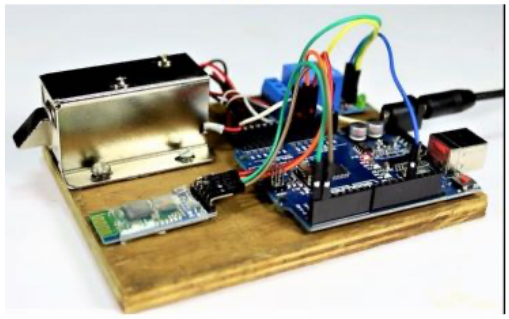


Figure 3

6.1 Hardware Unlock Experimental Output

- The Android app would also need to be programmed to communicate with the IoT device and display the lock status and control options to the user. This could involve using APIs or SDKs provided by the IoT device manufacturer, as well as developing custom code for the app.

- Overall, the detailed results of implementing a smart lock using IoT and Android would depend on the specific hardware and software technologies used, as well as the programming and design skills of the development team.

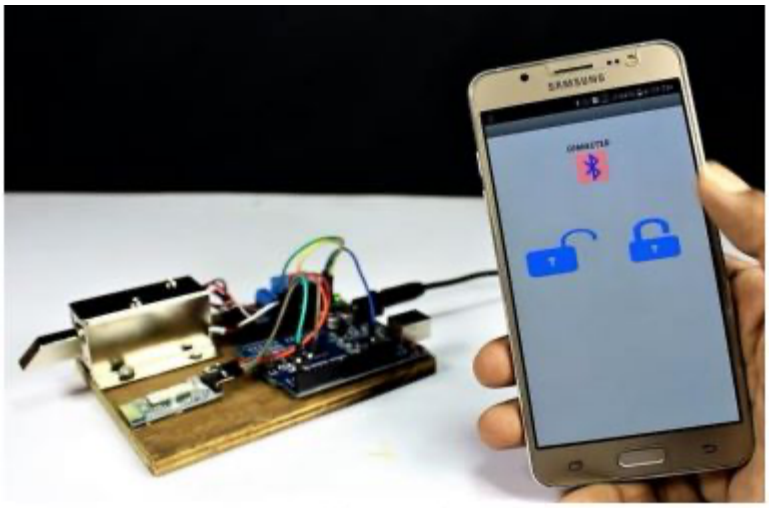


Figure 4

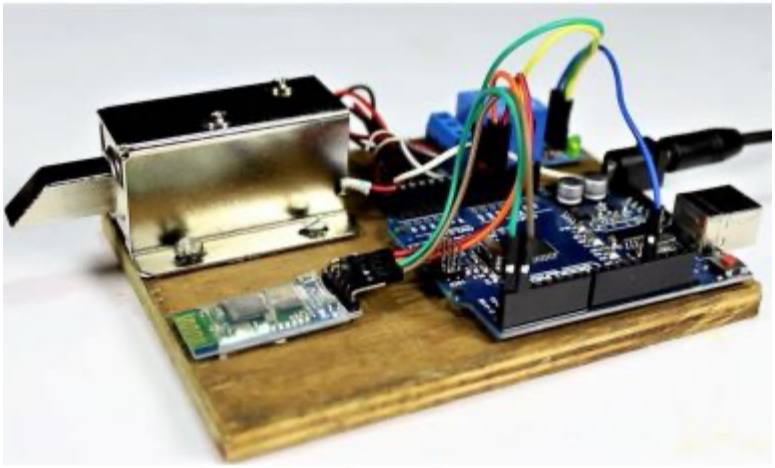


Figure 5

6.2 Hardware Lock Experimental Output

6.2 The Android Application Results :

The Below are the Android Applications results for the Application developed over the Android SDK which connects to the relay Module in The IOT lock system for Locking and Unlocking of the Lock.

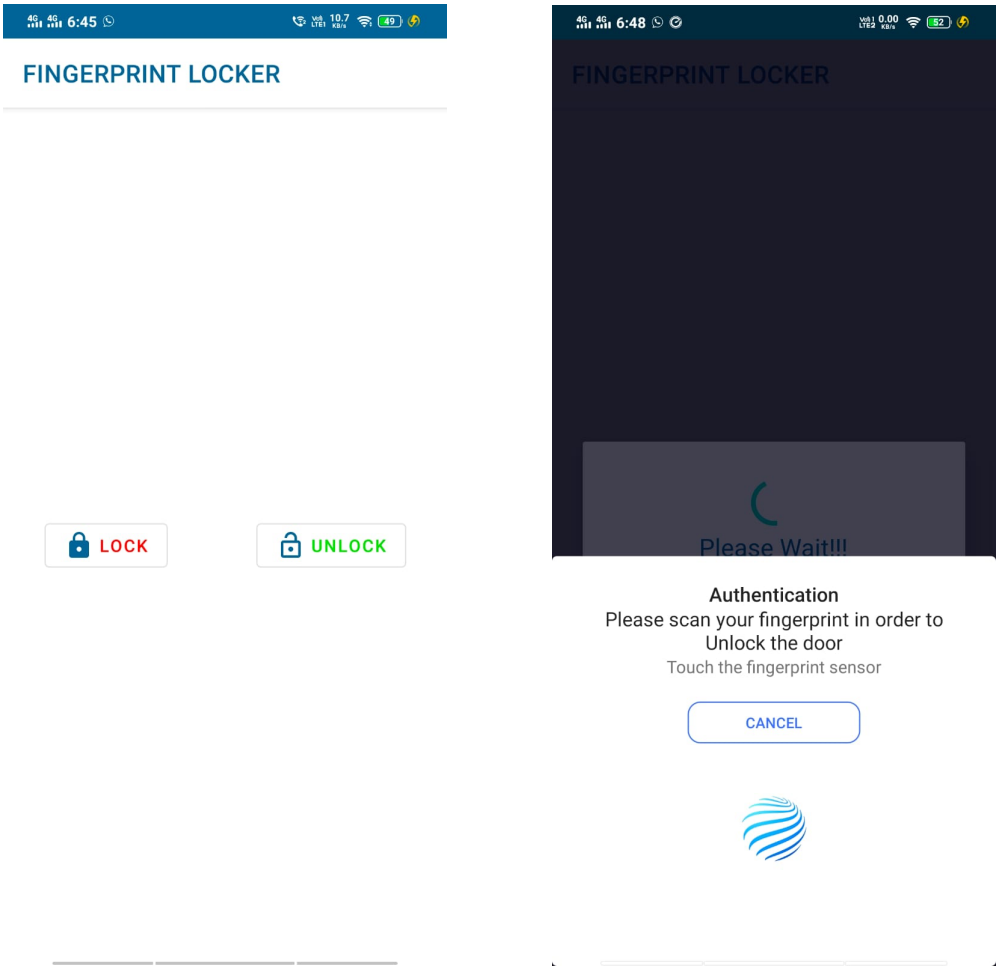


Figure 6.3 Android Experimental Output

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

This chapter provides the conclusions drawn after the implementation of the project.

7.1 CONCLUSION

The main aim of project is to design a door lock system using ARDUINO and Bluetooth applications. So the people can feel safe about their home or workplace whether they are inside or away from their place. The project is based on ARDUINO NANO and coding is done on ARDUINO IDE Open Software Platform. At the end, we achieved aim and objectives of this project. People can feel more secure about their home and workplaces. Door locking can be controlled conventionally with those of access. It is safe to say that main aim and objective of this project has been achieved finally.

The plan and usage of a unique mark-based lock framework is adjustable and adaptable. This entryway locking instrument is similarly practical than the accessible lock frameworks in the customary market. In our nation, private and government associations are a lot worried about security. Numerous organizations are keen on utilizing this kind of locking instrument yet the framework which is accessible has extremely high establishment cost. Because of this exorbitant cost, numerous little firms can't bear the cost of such frameworks.

7.2 FUTURE ENHANCEMENTS

The developments that can be made on the current implementation is as stated below:

During the project, some of problems came across the project is door to close automatically after a while of time it is open until that we can close from android application. For a future purpose that to make doors close automatically after certain time of delay or certain time of not using. The door and system is accessed for certain range of distance until we can connect Bluetooth for future purpose we can see into the access for longer distances.

Biometric technologies are rapidly becoming part of daily life. The future of biometric trends is medicine, banking, marketing research and in many other industries. Personal identification is used in all these mentioned places for security purpose. Biometric attendance system using Arduino uno is very useful for many industries and offices. It's easy, cost effective and works very well. Hence the future scope of this technology is widespread and quite essential in both domestic and industrial applications.

BIBLIOGRAPHY

1. [1] G. Fortino, A. Guerrieri and W. Russo, Middlewares for Smart Objects and Smart Environments: Overview and Comparison, Springer, Cham, 2014.
2. [2] C.-L. Hsu and J. C.-C. Lin, "An empirical examination of consumer adoption of Internet of Things services: Network externalities and concern for information privacy perspectives," Computers in Human Behavior, vol. 62, 2016.
3. [3] L. Adler, "Data-Smart City Solutions," Harvard Kennedy School, 18 February 2016. [Online]. Available: https://datasmart.ash.harvard.edu/news/article/how-smart-citybarcelona-brought-the-internet-of-things-to-life-789. [Accessed 03 April 2019].
4. [4] L. Columbus, "Forbes," Forbes, 6 June 2018. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2018/06/06/10-charts-that-will-challenge-yourperspective-of-iots-growth/#7b0e28d33ecc. [Accessed 3 April 2019].
5. [5] T. P. Norman, Electronic access control, Amsterdam: Butterworth-Heinemann, 2012.
6. [6] J. Stern, "iPhone X Review: How We Tested (and Tricked) FaceID," Wall Street Journal, 31 October 2017. [Online]. Available: https://www.wsj.com/articles/iphone-x-how-wetested-and-tricked-faceid-1509465766. [Accessed 4 April 2019].
7. [7] J. R. Delaney, "The Best Smart Locks for 2019," PCMag, 15 April 2019. [Online]. Available: https://www.pcmag.com/article/344336/the-best-smart-locks. [Accessed 18 April 2019].
8. [8] "August," August, [Online]. Available: https://august.com. [Accessed 3 April 2019].
9. [9] "Kwikset," Kwikset, [Online]. Available: https://www.kwikset.com. [Accessed 3 April 2019].
10. [10] B. Bruegge and A. H. Dutoit, Object-Oriented Software Engineering Using UML, Patterns, and Java, Pearson, 2009.
11. [11] M. Zyla, C. Dickson-Burke, D. Kim and A. Karet, "Developing a Mobile Application to Reduce Risks for Runaway Youth in Bangkok, Thailand," March 2018. [Online]. Available: https://digitalcommons.wpi.edu/iqp-all/2232/. [Accessed 3 April 2019].
12. [12] "The Good and the Bad of ReactJS and React Native," altexsoft, 10 September 2018. [Online]. Available: https://www.altexsoft.com/blog/engineering/the-good-and-

the-bad-ofreactjs-and-react-native/#pros%20of%20react%20native. [Accessed 3 April 2019].

13. [13] "Differences Between Raspberry Pi 3 vs Arduino," EDUCBA, [Online]. Available: https://www.educba.com/raspberry-pi-3-vs-arduino/. [Accessed 3 April 2019].
14. [14] "What is an Ultrasonic Sensor?" Keyence, [Online]. Available: https://www.keyence.com/ss/products/sensor/sensorbasics/ultrasonic/info/. [Accessed 3 April 2019].
15. [15] "Passive infrared sensor," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Passive_infrared_sensor. [Accessed 20 March 2019].
16. [16] "What's the Difference Between Pneumatic, Hydraulic, and Electrical Actuators?" MachineDesign, 16 April 2015. [Online]. Available: https://www.machinedesign.com/linear- 32 motions/what-s-difference-between-pneumatic-hydraulic-and-electrical-actuators. [Accessed 15 April 2019].
17. [17] "UML 2 Class Diagrams: An Agile Introduction," AgileModeling, [Online]. Available: http://agilemodeling.com/artifacts/classDiagram.htm. [Accessed 20 March 2019].
18. [18] "UML 2 State Machine Diagrams: An Agile Introduction," Agile Modeling, [Online]. Available: http://agilemodeling.com/artifacts/stateMachineDiagram.htm. [Accessed 20 March 2019].
19. [19] "Azure App Service," Microsoft, [Online]. Available: https://azure.microsoft.com/enus/services/app-service/. [Accessed 20 March 2019]

DrillBit

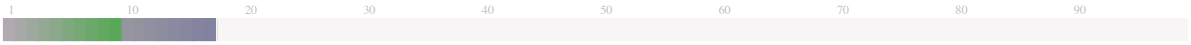
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

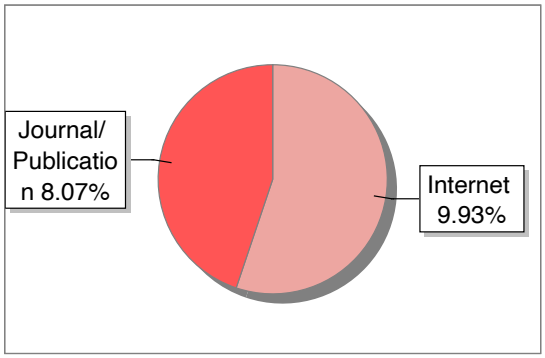
Author Name	Batch5
Title	Smart Unlock Using Fingerprint”
Paper/Submission ID	736047
Submission Date	2023-05-08 15:06:58
Total Pages	57
Document type	Project Work

Result Information

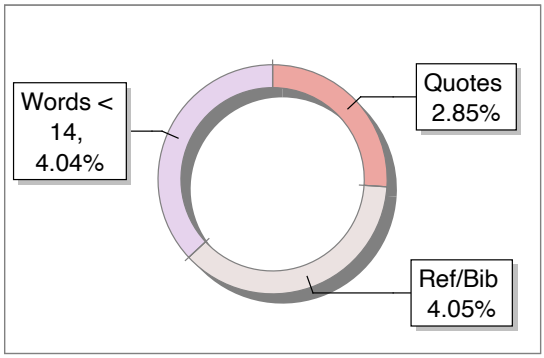
Similarity **18 %**



Sources Type



Report Content



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

18	45	B
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	web.wpi.edu	<1	Publication
2	web.wpi.edu	3	Publication
3	web.wpi.edu	2	Publication
4	moam.info	2	Internet Data
5	moam.info	1	Internet Data
6	en.wikipedia.org	1	Internet Data
7	www.ijert.org	1	Internet Data
8	ijariie.com	1	Publication
9	newwave.vn	1	Internet Data
10	sawasdeemedia.com	1	Internet Data
11	doku.pub	<1	Internet Data
12	SeCoMan A Semantic-Aware Policy Framework for Developing Privacy-Pres by Huerta-2014	<1	Publication
13	worldwidescience.org	<1	Internet Data
14	index-of.es	<1	Publication

NEW HORIZON COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC

Accredited by NAAC with 'A' Grade

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
CERTIFICATE ON PLAGIARISM CHECK

1	Name of the Student	Aditya Sunit Kanoi Hast Sinha Gana Sasank Reddy Harshvardhan Kumar Jaiswal
2	USN	1NH19IS009 , 1NH19IS055 , 1NH19IS194 , 1NH19IS195
3	Course	UG
4	Department	ISE
5	Mini Project/Project Report / Internship Report/Seminar Report/ Paper Publication/Ph.D. Thesis	Project Report
6	Title of the Document	NO KEY SMART UNLOCK USING FINGERPRINT
7	Similar Content(%)identified	18%
8	Acceptable maximum limit (%) of similarity	25%
9	Date of Verification	6TH MAY 2023
10	Checked by (Name with signature)	Guide Name : Dr. Sivaramankrishnan Signature :
11	Specific remarks, if any :	☑ 1st ☐ 2nd Attempt (Tick anyone)

We have verified the contents as summarized above and certified that the statement made above are true to the best of our knowledge and belief.

Research coordinator

ICACCS 2023

Event by & at :

Sri Eshwar College of Engineering
An Autonomous Institution
Coimbatore, India

9th 2023 International Conference on
Advanced Computing & Communication Systems

TECHNICAL SPONSORS IEEE IEEE Madras Section

Certificate of Presentation

Certify that

B. Swathi

New Horizon College of Engineering, Bangalore, India.

has presented a paper in the International Conference on

Advanced Computing & Communication Systems - ICACCS 2023

on 17th & 18th March 2023 at Sri Eshwar College of Engineering, Coimbatore, TamilNadu, India.

Paper Title :

No Key - Smart Door Unlock System using Fingerprint from Bluetooth Device

Dr. H. Anandakumar
Conference Chair

Dr. R. Subha
Convener

Dr. Sudha Mohanram
Patron

ICACCS

ICACCS 2023

Event by & at :

Sri Eshwar College of Engineering
An Autonomous Institution
Coimbatore, India

9th 2023 International Conference on
Advanced Computing & Communication Systems

TECHNICAL SPONSORS IEEE IEEE Madras Section

Certificate of Presentation

Certify that

Aditya Sunit Kanoi

New Horizon College of Engineering, Bangalore, India.

has presented a paper in the International Conference on

Advanced Computing & Communication Systems - ICACCS 2023

on 17th & 18th March 2023 at Sri Eshwar College of Engineering, Coimbatore, TamilNadu, India.

Paper Title :

No Key - Smart Door Unlock System using Fingerprint from Bluetooth Device

Dr. H. Anandakumar	Dr. R. Subha	Dr. Sudha Mohanram
Conference Chair	Convener	Patron

ICACCS 2023

Event by & at :

Sri Eshwar College of Engineering
An Autonomous Institution
Coimbatore, India

9th 2023 International Conference on
Advanced Computing & Communication Systems

TECHNICAL SPONSORS IEEE IEEE Madras Section

Certificate of Presentation

Certify that

Gana Sasank Reddy Gajjala

New Horizon College of Engineering, Bangalore, India.

has presented a paper in the International Conference on

Advanced Computing & Communication Systems - ICACCS 2023

on 17th & 18th March 2023 at Sri Eshwar College of Engineering, Coimbatore, TamilNadu, India.

Paper Title :

No Key - Smart Door Unlock System using Fingerprint from Bluetooth Device

Dr. H. Anandakumar	Dr. R. Subha	Dr. Sudha Mohanram
Conference Chair	Convener	Patron

ICACCS 2023

Event by & at :

Sri Eshwar College of Engineering
An Autonomous Institution
Coimbatore, India

9th 2023 International Conference on
Advanced Computing & Communication Systems

TECHNICAL SPONSORS IEEE IEEE Madras Section

Certificate of Presentation

Certify that

Harshvardhan Kumar Jaiswal

New Horizon College of Engineering, Bangalore, India.

has presented a paper in the International Conference on

Advanced Computing & Communication Systems - ICACCS 2023

on 17th & 18th March 2023 at Sri Eshwar College of Engineering, Coimbatore, TamilNadu, India.

Paper Title :

No Key - Smart Door Unlock System using Fingerprint from Bluetooth Device

ICACCS

Dr. H. Anandakumar	Dr. R. Subha	Dr. Sudha Mohanram
Conference Chair	Convener	Patron

ICACCS 2023

Event by & at :

Sri Eshwar College of Engineering
An Autonomous Institution
Coimbatore, India

9th 2023 International Conference on
Advanced Computing & Communication Systems

TECHNICAL SPONSORS IEEE IEEE Madras Section

Certificate of Presentation

Certify that

Hast Sinha

New Horizon College of Engineering, Bangalore, India.

has presented a paper in the International Conference on

Advanced Computing & Communication Systems - ICACCS 2023

on 17th & 18th March 2023 at Sri Eshwar College of Engineering, Coimbatore, TamilNadu, India.

Paper Title :

No Key - Smart Door Unlock System using Fingerprint from Bluetooth Device

ICACCS

Dr. H. Anandakumar	Dr. R. Subha	Dr. Sudha Mohanram
Conference Chair	Convener	Patron

No Key - Smart Door Unlock System using Fingerprint from Bluetooth Device

Dr.B.Swathi
Sr.Assistant Professor
Information Science and Engineering
New Horizon College of Engineering, Bangalore

Aditya Sunit Kanoi [1]
Information Science and Engineering
New Horizon College of Engineering, Bangalore

Gana Sasank Reddy Gajjala [2]
Information Science and Engineering
New Horizon College of Engineering, Bangalore

Harshvardhan Kumar Jaiswal [3]
Information Science and Engineering
New Horizon College of Engineering, Bangalore

Hast Sinha [4]
Information Science and Engineering
New Horizon College of Engineering, Bangalore

Abstract —

Technology has improved, and smart locking systems have become more sophisticated. In this case, the android-based Smart System is primarily intended for multimode operations. Such a system is necessary in banks and businesses since it provides functions that let users control locks. The implementation's efficiency the system is incredibly helpful because of its functionality and user-friendly interface. Some homeowners aim to connect their home's numerous home automation devices. Those connected to a Windows-based PC are the most popular home controllers. In our study, we introduced a form of smart technology that utilized Bluetooth while using a mobile smartphone. Consequently, using it will be simpler and more effective.

Additionally, it supported the free and open-source Android and Arduino platforms. This paper proposes a door lock automation system that uses an Android smartphone with Bluetooth as the first piece of hardware. Following a description of the design and software development process, a Bluetooth-based Smartphone application for locking and unlocking doors is demonstrated. The task module acts as the agent in the hardware design for the door-lock system, the Arduino microcontroller serves as the controller and data processing hub, and the solenoid acts as the door lock output. The results of each test show that it is compatible with the original plan for this study.

Keywords:

1) Bluetooth Door Lock

2) Biometrics

3) Internet Ignored Authentication

4) Un-Hackable Bluetooth Paired

5) Home security

I. INTRODUCTION

In spite of abundant benefits of [1-3] biometrics-based private confirmation structures over traditional freedom orders established tokens or information, they are susceptible to attacks that can decrease their protection considerably. Biometrics-located individual authentication scheme that use corporeal (mark on finger, face) or behavioral (talk, manuscript) characteristics are becoming more and more well-known, distinguished to traditional holes that are established tokens (key) or information(password).

Fingerprint-located labelling is individual of the most main biometric sciences that have drawn a solid amount of consideration currently. Fingerprint technology is so low in individual identification that it has existed well settled. Each human has singular owns mark, even the twin has various marks on finger. So, fingerprint acknowledgment is a valuable instability law request.

The [4-6]photoelectric lock utilizing fingerprint acknowledgment includes a process of proving the user's similarity by utilizing mark on finger recognition as a key to the photoelectric lock. This work climaxes the happening of fingerprint acknowledgment plans using ARDUINO 1.6.3. To perceive the recommendation mark on finger image from the stocked samples in bmp, tif; argument; jpg;jpeg; gif file type. Then the facts of the recognized dab countenance will be stored in a table for verification approved by the consumer. These marks on finger recognition orders establish the theory that the human fingerprint is singular. It is mainly to confirm the individuality of dab in consideration of using the dab image for protection

II. LITERATURE SURVEY

Paper-1 :Wireless Biometric Lock using Arduino with the IOT: Published by Prof. Sumedh V Dhole, Akshay Kumar, Mayank Gupta and Rishabh Arora. The paper was published in 2020. High security and assurance. Biometric identification provides the answers to something a person has and helps verify identity. Convenient and fast. Non-transferrable.

Paper-2 :Short term Fourier transforms. Contextual filtering in Fourier domain: Published by Sherlock. Published in 1994. Fingerprint image can be computed using single unified approach. More formal approach for analyzing the non-stationary fingerprint image.

Paper-3 : Log Gabor Filter: :Published by Wang, Li Huang, Feng . Overcomes the drawbacks of traditional Gabor filter, Improves fingerprint enhancement performance.

Paper-4 :Smart Door Unlock System Using Fingerprint. Published by K.Rajesh, B.Venkata Rao, P.AV.S.K.Chaitanya, A.ruchitha Reddy. Explains the overall perspective of use of IOT in Fingerprint technology and using of the Arduino nano System for Integration.

III. EXISTING SYSTEM

Most doors in the current system are managed by individuals using Keys, security cards, and a password to enter locked doors. [7-9]The development of wireless controlling technology is a result of this. With a skillful fusion of modern technology and embedded systems, purpose is achieved. However, we're planning to make a modification that will make things more secure when compared to the current system the existing system used regular, conventional locks that have been around since quite previous times Traditional locks do have a drawback, though, in that they may be readily broken by burglars, or they can go missing. Then we upgraded to the modern locks that can be used with pin numbers, passwords, and other security measures. The drawback of these locks is that we can forget our passwords or pin numbers. We also have biometrics that can be used to open the door lock itself, but we are now proposing a new style of modern wireless door lock utilizing the same biometric.

IV. PROPOSED SYSTEM

We are developing a door lock that can be opened using biometrics in the proposed system. These days, it is fashionable for them to be secure and simple for the owner or administrator to open. We used a biometric door lock to create this, however a cell phone will be used to unlock the lock.

We are developing an app that will enable mobile phone door lock access. Can be connected through Bluetooth. To build an app, we're using Kotlin, and we've given Bluetooth access to operate the door lock. Using Android Studio, we are developing a kotlin application in which we will use the kotlin programming language. Kotlin was used to create the code to communicate with the [10-13]HC-05 Bluetooth module.

To connect with the HC-05 module, a basic Bluetooth adapter and a Bluetooth socket (integrated within the software) were utilized. Additionally, this connection allowed data to be sent across the socket from the [14] Rx and Tx pins of the Arduino Nano. The software instantly establishes a connection with the HC-05 Board, and the device's fingerprint scanner serves as an authentication factor.

V. METHODOLOGY

Through Arduino, we must provide a power source with an appropriate voltage. Open the app that was developed using the Kotlin application after providing electricity. The mobile device must turn on for Bluetooth and connected to the HC-05 Bluetooth Module before the Bluetooth symbol in an app change to a lock icon . Next, we must tap the fingerprint symbol . It will notify us that the fingerprint is required to unlock it . Keep your finger on the fingerprint reader on your phone right now. If the fingerprint is recognized by our phone, the lock is turned on and the lock icon is changed to an unlock icon.

Elements used:

- Bluetooth module HC-05
- Arduino Nano
- The Solenoid lock
- Relay module
- Connecting Wires
- Battery

Connect the HC-05 module's Tx and Rx pins to the Arduino's Rx and Tx pins, respectively. With the Arduino mini, align the ground and vcc pins. Connect the Arduino's D9 pin to the Relay modules "in" input. Connect the relay and the solenoid lock to the 12-volt battery system. With the aid of the app, the relay, which functions as a switch, will be changing its state.

A. Bluetooth Model HC-05:

A Bluetooth module called HC-05 is created for wireless communication. This module may be set up as either a grasp or a slave. The HC-05 contains a red LED that indicates the connection status, including whether or not Bluetooth is involved. Before being connected to the HC-05 module, this red LED constantly pulses periodically.

[16]Its blinking reduces to two seconds when it is linked to another Bluetooth device. The 3.3 V needed for this module. We can also connect a 5V supply voltage because the module includes a built-in 5 to 3.3 V regulator. There is no need to change the HC-05 Bluetooth module's transmit level because it contains a V stage for RX/TX and the microcontroller can sense 3.3 V levels.

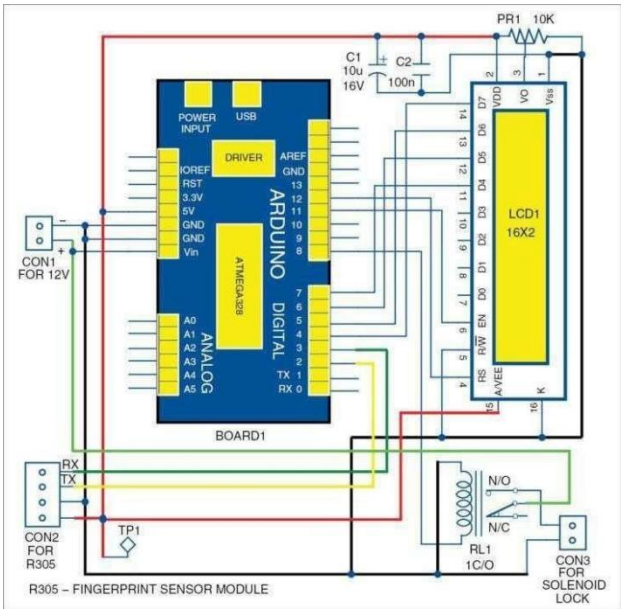
B. Ardunio Nano:

Arduino nano is an open-source tool for prototyping electronic devices that relies on flexible, easy-to-use hardware and software. For artisans, architects, and professionals, and everyone was fascinated by creating clever articles or situations. It's a large figuring step based on an open source

microcontroller board, and advancement environment for writing board programming programmers. Simply put I n other terms, Arduino is a tiny microcontroller board with a USB connector to connect to your computer and some networking add-ons that could be connected to external devices, such as engines, transfers, light sensors, laser diodes, amplifiers, mouthpieces, etc.

Both may be powered by a 9V battery or by a USB connection from a computer.

They can be managed or changed through a computer, then detached and given free rein to operate. Compared to their ancestors, which were designed to cover greater area and were thus more expensive and capable of performing fewer features, electronic gadgets are becoming increasingly compact, flexible, and lightweight.



C. The Selenoid lock:

It is available in opening in the force-on mode type as well as bolting and protecting in this type, which may be used individually for circumstances. Only when the solenoid is powered on is opening possible with the strength-on opening type. When an entrance of this kind is guarded, it won't open in the event of an energy outage or twine separation, providing excellent assurance. This kind is used specifically in locations where it is necessary to prevent misconduct. When the solenoid is turned on, the strength-on locking type can bolt a door. The entrance is made available if the strength is severed. [17]This kind opens the door in the case of a string separation caused by a fire or an unforeseen incident, and it is used for emergency exits where extinguishing fires, leaving, or making specific arrangements for departure are required instead of providing assurance for wrongful counteraction. The keeping type maintains the no-power country in every job while doing two tasks—bolting and opening—by applying a good or bad heartbeat voltage to the solenoid.

This type of capability can save energy because it is simple to turn the solenoid on continuously. The continuous rating is intended to maintain an assessed voltage power continuously for a significant amount of time without exceeding a targeted temperature upper push limit, whereas the irregular rating is intended as a method to maintain a specified voltage with ease for a predetermined amount of time without going over a specific temperature upper limit.

D. Relay module:

An electromechanical device known as a relay can be used to create or sabotage an electrical connection. In essence, a relay is just like a mechanical change, except that you may control it with a digital sign rather than alternately turning it

on and off. It consists of a flexible transferable mechanical phase that can be managed electronically using an electromagnet. Once more, this relay functioning principle is only appropriate for electromechanical relays.

Therefore, a relay is a switch that electromechanically regulates (opens and closes) circuits. The main function of this device is to create or disrupt contact with the assist of an assign without the need for human intervention to turn it ON

or OFF. It is mostly used to control an overpowered circuit by applying a weak electrical signal. Typically, a DC signal is used to operate a circuit that is driven by high voltage, such as controlling AC household appliances using DC signals from microcontrollers.

E. Connecting Wires:

In contemporary connections and energy links, artificial polymer protection that resembles elastic is employed. Because of its chief moisture hindrance, it was built underground.

The most severe working temperature at the conductor surface and the recommended working voltage are used to evaluate protected connections. A link may also provide several program usage scores, such as one rating for dry venues and another when exposed with moisture or oil.

F. Battery :

A battery is a tool that is constructed of one or more electrochemical cells with external connections and is used to power electrical devices including mobile phones, flashlights, and electric vehicles. When a battery is connected to an external electric load, a [18] redox reaction transforms the high energy reactants. The battery's good terminal is the cathode, while its bad terminal is the anode. The terminal marked "poor" is the supply of the electrons that will drift via an external electric-fueled circuit to the tremendous terminal.

Regarding the low-energy products, And the Free Power Distinction Is Brought to External Circuit Known As Electricity Clearly The Term "Battery" Specifically Referred To Device Made Out of More Than One Cell, But The Usage Is Developed To Include Devices Made Of An SingleCellular.

APPLICATIONS

1} They are playing a very important role in many disciplines today.

2} This is used in smart homes across multiple cities.

3} Used in Defense and Security for advanced secure and protected system.

4} These locks are used in industries and modern warehouses.

5} Used in medical and research centers to keep high end database hardware secure.

FUTURE SCOPE

While doing the project, we faced problems with the project's door closing automatically after a while of time it was open until we could close it from the android application. For a future purpose we are planning to create an application such that the doors will be closed automatically after some seconds of not using the door or directly through the application.

In the current application we are planning to access the system within a certain distance which would not be that much long as it might be expected because of the Bluetooth connectivity it can be only accessed in that specific range of Bluetooth.

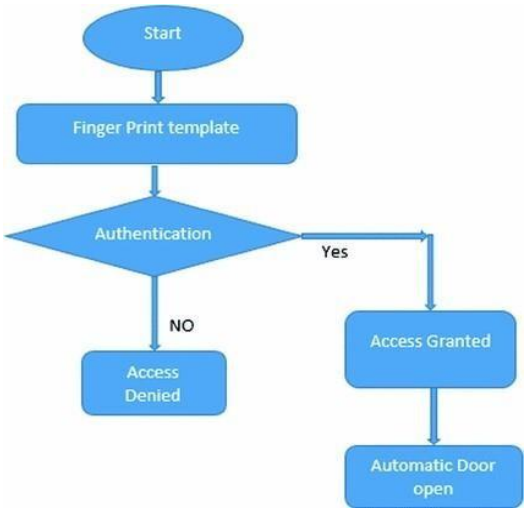
RESULTS

The program for ARDUINO NANO was written in C++ language. It was then compiled using ARDUINO IDE. After that code is dumped or uploaded into a microcontroller. We have created android app by using KOTLIN Java Language. After that we are connecting the hardware after giving power supply. [19-21]The android app communicating with ARDUINO NANO using wireless Bluetooth communication. If the communication happens in a correct way, the door lock opens and closes. The communication and result flow below using flow chart.

DISCUSSION

Project gives idea how to control door locks. Carbon paper locking (Password, iris detection, RFID) is used as prototype for indoor and outdoor locking system. This project is based on smart phone and ARDUINO platform both are free open source. So the implementation of project is inexpensive and it can hold reasonable for common person now these days.

The system is designed and prototyped to control the condition of door locking using Bluetooth enabled smart phone and Bluetooth communication through HC-05 Bluetooth module. A simple prototype is discussed for this project.



Fingerprint Access Verification

Conclusion

Fingerprint labeling embellishes the safety of a Door and makes it likely only for a few picked nations to use the Door. Thus, by achieving this comparably inexpensive and surely free scheme on a Door, individuals can guarantee much better safety and uniqueness than that offered by a normal lock and key. It may be understood that the use of biometric safety orders offer a much better and fool proof wealth of confining the use of Doors by unjustified consumers. The grown prototype serves as a force to drive future research, equip towards evolving a stronger and more entrenched real-opportunity mark.

References

[1] Kavita Gupta "ARDUINO BASED SMART IoT Palm print Recognition" International Journal of Engineering Applied Sciences and Technology, 2020 Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 592-597 Published Online April 2020 in IJEAST (http://www.ijeast.com)

[2] Debnath Bhattacharyya,Rahul Ranjan, "Biometric Authentication : A Review" International Journal of u- and e-Service Science and Technology,Volume-3,Issue-5, September 2009.

[3] S. Prabhakar, S. Pankanti, A. K. Jain, "Biometric Recognition: Security and Privacy Concerns", IEEE Security &Privacy, MarcWApril2003, pp. 33-42.

[4] D. Maltoni, D. Cappelli, R. Wayman, J.L& Jain "Fingerprint Verification Competition" IEEE Transactions on Pattern Analysis and Machine Intelligence, A. K. (2002). FVC2000, Vol. 24, No. 3, pp. 402-412

[5] Baswaraju Swathi, Abhishek Kumar, Ishu Kumar, Vathsavi Venkat. Implementation of Improved Billing System. International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2020 IJSRCSEIT | Volume 6 | Issue 3 |ISSN: 2456-3307.

[6]J. Feng, A. K. Jain, A. Ross, "Fingerprint Alteration", MSU Technical Report, MSU-CSE-09-30, Dec. 2009 "

[7] Ketan Rathod, Rambabu Vatti, Mandar Nandre and Sanket Yenare, "Smart door security using Arduino and Bluetooth application", International Journal of Current Engineering and Scientific Research, vol. 4, no. 11, pp. 73-77, 2017

[8] O. Agbo David, Madukwe Chinaza and O. OdinyaJotham, "Design And Implementation Of A Door Locking System Using Android App", International Journal Of Scientific & Technology Research, vol. 6, no. 08, pp. 198-203, 2017.

[9] Mounica, B., & Merlin Mathew, & Aishwarya Balakrishnan, & Bikky Kumar Goit, (2019). Learning Dynamic Toll Charge using openCV and Spark. International Journal of Scientific Research in Computer Science, 3(5), (33-37).

[10] Karen Scarfone and John Padgette, Guide to Bluetooth Security. Special Publication 800–121 Recommendations of the National Institute of Standards and Technology, 2008

[11] G.V. Zaruba and I. Accelerating, "Bluetooth inquiry for personal area networks".

[12] S. Krco, "Bluetooth based wireless sensor networks implementation issues and solutions", 10 th Telecommunications forum (TELEFOR 2002) , NOV. 2002

[13] S. Rajkiran and C. Balakrishnan, "RSSI estimation using Bluetooth in smart phones", 2015 2nd International Conference on Electronics and Communication Systems (ICECS), pp. 1431-1434, 2015..

[14] S.B. Sagarkumar, P. Chatur and B.V. Wakode, "Data Logger System: A Survey", *National Conference on Emerging Trends in Computer Science & Information Technology, pp. 24-26, 2011..*

[15] Baswaraju Swathi and Harshvardhan Tiwari, "Integrated Pairwise Testing based Genetic Algorithm for Test Optimization" International Journal of Advanced Computer Science and Applications (IJACSA), 12(4), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120419.

[16] Baswaraju Swathi, Harshvardhan Tiwari. Test Case Generation Process using Soft Computing Techniques. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-1, November 2019.

[17]K. S. Kow, J. k. Yang, D.H. Hwang, H.S. Ko, C.O. Ga and J.P. Cho, "Building Fire Prediction Model Study Using AI", The Journal of Korean Institute of Communications and Information Sciences 20-07, vol. 45, no. 07, pp. 1210-1218, 2020

[18] Shobha, T., Anandhi R.J. Adaptive strategy operators based GA for rule discovery. Int. j. inf. tecnol. 12, 1365–1375 (2020). https://doi.org/10.1007/s41870-019-00303-z.

[19] B. Swathi, "Automated Test Case Prioritization and Evaluation using Genetic Algorithm," 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), 2022, pp. 1-5, doi: 10.1109/IC3SIS54991.2022.9885535.

[20] B. Swathi, P. T, P. K. S, S. Y and G. V, "Survey on IoT based Farm Freshness Mobile Application," 2022 International Conference on Advanced Computing Technologies and Applications (ICACTA), 2022, pp. 1-7, doi: 10.1109/ICACTA54488.2022.9752796.