SMART DOOR LOCK SYSTEM

A

MAJOR PROJECT-II REPORT

Submitted in partial fulfillment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GROUPNO.39

Jividha Koshti	**0187CS211080**
Manika Shrivastava	**0187CS211094**
Hamza Ul Hasan	**0187CS211069**
Ritika Dubey	**0187CS211137**

Under the guidance of

Prof. Amit Swami

(Assistant Professor)

Department of Computer Science &Engineering
Sagar Institute of Science & Technology (SISTec), Bhopal(M. P)

Approved by AICTE, New Delhi & Govt.of M. P.
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M. P.)

June-2025

Sagar Institute of Science & Technology (SISTec), Bhopal (M.P.)

Department of Computer Science & Engineering

CERTIFICATE

We hereby certify that the work which is being presented in the B. Tech. Major Project-II Report entitled **SMART DOOR LOCK SYSTEM,** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology,*** submitted to the Department of **Computer Science & Engineering**, Sagar Institute of Science & Technology(SISTec)**,** Bhopal(M.P.)is an authentic record of our own work carried out during the period from Jan-2025 to June-2025 under the supervision of **Prof. Amit Swami.**

The content presened in this project has not been submitted by me for the award of any other degree elsewhere.

Jividha Koshti	**Manika Shrivastava**	**Hamza Ul Hasan**	**Ritika Dubey**
0187CS211080	**0187CS211094**	**0187CS211069**	**0187CS211095**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Prof. Amit Swami	**Dr. Amit Kumar Mishra**	**Dr. D. K. Rajoriya**
Project Guide	**HOD,CSE**	**Principal**

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. D. K. Rajoriya**, **Principal, SISTec and Dr. Swati Saxena, Vice Principal SISTec** Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We also take this opportunity to express a deep sense of gratitude to **Dr. Amit Kumar Mishra, HOD, Department of Computer Science & Engineering** for his kindhearted support.

We extend our sincere and heartfelt thanks to our guide, **Prof. Amit Swami,** for providing us with the right guidance and advice at the crucial junctures and for showing us the right way.

I am thankful to the **Project Coordinator, Prof. Deepti Jain** who devoted her precious time in giving us the information about various aspects and gave support and guidance at every point of time.

I would like to thank all those people who helped me directly or indirectly to complete my project whenever I found myself in any issue.

TABLE OF CONTENTS

ABSTRACT

In today's world, security is a prime concern for homes, offices, and institutions. Traditional lock-and-key mechanisms are prone to security risks such as key duplication and unauthorized access. This project proposes an IoT-based Smart Door Lock System that enhances security through NodeMCU and the Blynk app for remote access control. The system allows authorized users to lock and unlock the door via a mobile application, eliminating the need for physical keys. Real-time status updates and access logs ensure secure monitoring. The Wi-Fi-enabled NodeMCU microcontroller connects with the Blynk cloud to process authentication requests, granting access based on predefined permissions. This solution provides keyless entry, remote monitoring, and enhanced security, making it an efficient and user-friendly alternative to conventional.

LIST OF ABBREVIATIONS

ACRONYM	FULLFORM
SDLC	Software Development Life Cycle
AI	Arduino IDE
PS	Power Supply
RM	Relay Module

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRDUCTION

The advancement of smart technologies has transformed traditional security systems into intelligent, automated solutions. This project, titled "IoT-Based Smart Door Lock System", focuses on enhancing home and office security by enabling keyless, remote-controlled access using the Internet of Things (IoT). The system is built using a NodeMCU (ESP8266) microcontroller, which connects to the Blynk mobile application via Wi-Fi, allowing users to lock or unlock the door remotely through their smart phones.

1.1ABOUT PROJECT

In today's fast-paced and digitally connected world, the importance of secure and intelligent access control systems has increased significantly. Traditional lock and key mechanisms, while simple, come with several limitations such as key misplacement, duplication risks, and lack of access tracking. The emergence of Internet of Things (IoT) has revolutionized how we interact with devices, allowing for smart solutions that offer better security, convenience, and control. One such advancement is the Smart Door Lock System, which allows users to manage door access using smart devices such as mobile phones.

This project focuses on the design and development of an IoT-based Smart Door Lock System using NodeMCU (ESP8266) and the Blynk mobile application. The goal is to create a cost-effective, user-friendly, and secure system that enables homeowners or administrators to remotely control and monitor the status of their door locks via an internet-connected device. The system integrates a relay module to control the door's locking mechanism and utilizes Wi-Fi connectivity for communication between the microcontroller and the mobile application.

By using the Blynk platform, users can interact with the system in real-time, receive notifications, and monitor access activity logs. This eliminates the need for traditional keys and offers a more secure alternative by ensuring that only authorized users can unlock or lock the door remotely. The system also supports the addition of optional security features such as RFID authentication, fingerprint modules, or keypad entry, making it highly customizable and scalable for future enhancements.

One of the key advantages of this system is its flexibility and accessibility. It is ideal for various applications, including residential homes, office buildings, hostels, and rental properties. Property owners can grant or revoke access to users without physical interaction. In rental or Airbnb-like setups, owners can offer temporary access credentials, improving convenience and safety for both the owner and the tenant.

Furthermore, the project addresses real-world challenges such as power efficiency, Wi-Fi dependency, and data privacy. Efforts are made to reduce energy consumption by using efficient microcontrollers and implementing features like sleep modes. To handle network unavailability, alternative access methods like RFID tags or local switches can be integrated to ensure that the system remains functional even without internet access. For added reliability, backup power supplies such as battery packs or UPS systems can be used to prevent system failure during power outages.

The use of open-source tools such as Arduino IDE for programming and Blynk for UI design makes the project affordable and easy to replicate. The system can be easily maintained and upgraded, making it a suitable solution even for small-scale deployment or academic demonstration purposes. The interface is designed to be minimal and intuitive, so users from all backgrounds can operate it comfortably without requiring technical knowledge.

With growing concerns about home security and automation, this project demonstrates how simple IoT technologies can be applied to solve everyday problems. It not only adds a layer of protection but also offers the convenience of remote access, making it suitable for the modern smart lifestyle.

In conclusion, the IoT-based Smart Door Lock System is a blend of security, automation, and user-friendly technology, providing a reliable and scalable solution to traditional door locking methods. This project highlights the potential of IoT in smart home systems and serves as a practical example of how engineering and technology can work together to create smarter, safer environments .One such innovative application is the IoT-Based Smart Door Lock System, which brings automation and security together to revolutionize the way we manage.

In today’s digital era, the demand for smart and secure systems is rapidly increasing due to growing concerns over safety, convenience, and automation. Traditional door locking mechanisms, though widely used, come with several limitations such as lost keys, unauthorized duplication, and lack of access tracking. These limitations have highlighted the need for more intelligent and connected solutions that provide better control, monitoring, and ease of use. The advancement of the Internet of Things (IoT) has paved the way for the development of smart devices that can communicate, analyze, and act on data without human intervention. One such innovative application is the IoT-Based Smart Door Lock System, which brings automation and security together to revolutionize the way we manage access to our homes and buildings.

This project aims to develop a smart locking system that allows users to control door access remotely using a smart phone application (Blynk app). At the core of the system is a NodeMCU (ESP8266) microcontroller, which connects to a relay module and controls the electronic lock mechanism. The NodeMCU is connected to the internet via Wi-Fi and communicates with the Blynk app to receive user commands. With just a tap on their mobile screen, users can lock or unlock the door from anywhere in the world, making the system extremely convenient and efficient. The system also provides real-time status updates and can be extended to store access logs, send alerts for unauthorized attempts, and even support additional features like RFID, fingerprint, or keypad access.

The smart door lock system is particularly useful in scenarios where physical presence is not possible, such as remotely allowing guests or service personnel into a home, or managing multiple access points in an office environment. In addition to enhancing security, the system also contributes to better user experience and home automation. Designed to be low-cost, energy-efficient, and scalable, this project demonstrates a practical and impactful use IoT technologies in the field of smart security system.

1.2 PROJECT OBJECTIVES

The primary objective of this project is to design and implement a secure, reliable, and user-friendly IoT-based Smart Door Lock System that enables users to control door access remotely using their smart phones. This project aims to enhance traditional security methods by replacing physical keys with a digital solution that leverages the power of the Internet of Things (IoT). By integrating a NodeMCU (ESP8266) microcontroller with a relay module and connecting it to the Blynk mobile application, the system allows authorized users to lock or unlock the door from any location via the internet. In addition to remote control, the project seeks to offer real-time monitoring, alerts, and feedback to the user regarding door status and access events. The system is designed to be cost-effective, energy-efficient, and easily scalable, making it suitable for a variety of environments such as homes, offices, and hostels. Another key objective is to ensure that the system remains functional and accessible even in the absence of Wi-Fi by providing optional offline access methods like RFID cards or keypad input. Furthermore, the project aims to create a flexible platform that can be enhanced in the future with features such as biometric authentication, voice assistant integration, or GSM-based communication. Ultimately, the goal is to develop a smart access control system that provides both security and convenience through modern technology and automation.

The IoT-Based Smart Door Lock System is developed with the aim of providing a modern, intelligent alternative to traditional lock-and-key mechanisms by integrating the benefits of IoT technology into everyday security. The system seeks to offer a seamless and secure door access experience through remote control, real-time monitoring, and flexible user management. One of the key objectives is to eliminate the physical constraints of conventional locks—such as carrying keys, sharing them manually, or the risk of them being lost or duplicated—by replacing them with a cloud-connected, smartphone-controlled solution. The project is designed to be highly modular and adaptable, making it suitable for diverse settings like residences, offices, hostels, rental apartments, and shared spaces where secure, role-based access control is required.

A core goal is to ensure that the system is not only technically functional but also affordable, simple to deploy, and user-friendly, especially for non-technical users. The use of open-source platforms like NodeMCU and Blynk enables ease of development and cost-efficiency. Through the Blynk mobile app, users are able to remotely lock or unlock the door, monitor access activity, and receive instant alerts in case of unauthorized attempts or access failures. The system also aims to maintain functionality in offline conditions by integrating alternative authentication methods such as RFID tags, keypads, or even biometric systems in future enhancements.

In addition, the project emphasizes scalability and future readiness, allowing for the integration of advanced features like voice control, geo fencing, auto-locking based on location, and GSM-based SMS control. Security is another major focus, with the aim of creating a solution that reduces unauthorized access while increasing control and visibility over entry and exit activity. This system offers not just a locking mechanism, but a smart, connected access management solution that fits into the growing ecosystem of smart home

CHAPTER 2
SOFTWARE & HARDWARE REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS

Software Requirements (Developer):

- Arduino IDE
- Blynk App
- Wi Fi

Software Requirements (Client):

- Blynk App
- Internet Connection
- User Authentication Credentials

2.2 HARDWARE REQUIREMENTS

Hardware Requirements (Developer):

- Node MCU (ESP8266/ESP32)
- Relay Module
- Electromagnetic/Solenoid Lock
- Power Supply (5V/12V Adapter or Battery)
- Jump Wires & Breadboard
- Sensors (Optional)

Hardware Requirements (Client):

- Smart phone or Tablet
- Wi-Fi Networ

CHAPTER 3
PROBLEM DESCRIPTION

Traditional door locking systems rely heavily on physical keys, which pose risks such as loss, theft, duplication, and lack of remote accessibility. These systems offer no real-time monitoring or control, making them inconvenient and insecure in many modern scenarios. In shared spaces like hostels, rental homes, or offices, managing keys becomes even more challenging. There is a need for a smarter, more secure, and remotely accessible solution that can be controlled and monitored from anywhere. This project addresses the issue by developing an IoT-based smart door lock that provides remote access, real-time updates, and enhanced user convenience.

3.1 OVERVIEW

In the age of digital transformation, technology is evolving rapidly to make everyday life smarter, safer, and more efficient. Security and access control are among the most crucial areas where technological advancements can bring significant improvements. Traditional mechanical lock-and-key systems have been widely used for centuries but are now increasingly seen as outdated due to their limitations. Issues like lost or stolen keys, unauthorized duplication, no remote control, and the lack of access tracking have created the need for more advanced, intelligent solutions. In response to these challenges, this project proposes the development of an IoT-Based Smart Door Lock System, which combines convenience, control, and enhanced security using modern technologies.

The proposed system uses a NodeMCU (ESP8266) microcontroller, which connects to a Wi-Fi network and communicates with the Blynk mobile application, allowing users to control the door lock from anywhere in the world using their smart phones. The central component of the system is a relay module, which is used to control the electronic locking mechanism—usually an electric strike or solenoid lock. When the user sends a command through the Blynk app, the NodeMCU receives it over the internet and triggers the relay to either lock or unlock the door. This provides users with real-time control over their door, eliminating the need for physical keys.

The user interface (UI) is provided by the Blynk app, which is easy to set up and allows for buttons, status indicators, and push notifications to be displayed on a smartphone. This means that users can not only control the lock remotely but also receive immediate feedback regarding the status of the door—whether it's currently locked or unlocked. In addition, the system can log access attempts and provide data for further analysis or alerts in case of suspicious activity.

One of the major advantages of this system is its applicability in a wide range of environments. For residential use, homeowners can unlock doors for guests even when they are not at home. In office settings, managers can control and monitor entry points securely. For hostels and rental accommodations, owners can grant or revoke access remotely without needing to issue physical keys. The flexibility and functionality of the system make it ideal for places where shared or rotating access is needed.

This project also addresses a critical challenge in IoT systems—reliance on continuous internet connectivity. While the primary mode of operation depends on Wi-Fi, the system is designed with extensibility in mind. Features such as RFID card readers, keypad modules, or biometric sensors can be integrated to serve as backup access methods in the event of a network failure. Future improvements could also include GSM modules for SMS-based control, or Bluetooth for local device communication.

Another key goal of the project is to provide a low-cost and energy-efficient solution that can be deployed even in budget-constrained environments. Using open-source hardware like NodeMCU and free platforms like Blynk and Arduino IDE keeps development costs low, while power-saving features of the ESP8266 module help reduce energy consumption. The simplicity of installation and minimal maintenance requirements make the system practical for everyday users.

From a development perspective, the project also serves as a valuable learning platform in fields such as embedded systems, networking, IoT protocols, and mobile interface design. The modular architecture allows students or developers to expand and customize the system easily—adding voice commands using smart assistants like Alexa or Google Assistant, integrating cloud-based analytics, or enhancing security through encryption and multi-factor authentication.

To summarize, the IoT-Based Smart Door Lock System offers a modern and reliable solution to the limitations of traditional locking mechanisms. It emphasizes remote accessibility, enhanced security, ease of use, and scalability. With smart access control becoming a standard feature in modern homes and offices, this system demonstrates how IoT can be effectively used to build intelligent, user-centric, and secure environments. As smart technologies continue to grow in popularity, this project lays a strong foundation for developing future-ready smart security solutions.

CHAPTER 4
LITERATURE SURVEY

In the realm of smart home automation and security, researchers and developers have explored various methods to enhance door locking mechanisms by integrating modern technologies. Traditional mechanical locks have several limitations such as lost keys, unauthorized duplication, lack of remote access, and no access tracking. These limitations have led to the development of intelligent door lock systems using technologies like Bluetooth, GSM, RFID, biometrics, and Wi-Fi. This literature survey explores and reviews the relevant existing systems, highlighting their features, benefits, and limitations, and how they inspired and influenced the design of our proposed system.

Early research and projects focused on Bluetooth-based smart locks using modules like HC-05 or HC-06 paired with Arduino boards. These systems allowed users to unlock doors via Bluetooth commands from smart phones. While this provided a wireless solution and eliminated the need for keys, the major drawback was the short-range limitation, typically 10 meters, and the lack of internet-based remote access. This made Bluetooth-based systems suitable only for very local environments and not practical for truly remote control.

To overcome the limitations of short-range wireless technologies, researchers developed GSM-based door locking systems using SIM800/SIM900 modules. These systems enabled users to control door locks by sending SMS commands. The main advantage of this model was its functionality in areas with no Wi-Fi access, making it reliable for remote regions. However, these systems were often slow and lacked real-time control, had limited interface features, and required constant monitoring of mobile signals. Despite their limitations, GSM-based solutions inspired us to consider GSM as a future enhancement to support offline or emergency access when internet connectivity is not available.

Another widely adopted method is the RFID-based smart locking system, where users carry RFID tags and gain access by scanning them through an RFID reader. These systems are effective for local offline access control and are used in hostels, offices, and schools. However, they do not provide remote access and require physical presence at the door. While RFID technology is highly reliable, its lack of internet connectivity makes it insufficient for smart home applications. However, the idea of using RFID for offline backup access inspired us to consider RFID integration in our system to ensure usability during network failures.

More recently, with the evolution of IoT and microcontrollers like NodeMCU (ESP8266), several projects began integrating Wi-Fi connectivity to enable smart door locks to be controlled via cloud-based platforms and mobile apps. One popular solution used in many projects is the Blynk mobile app, which allows developers to create simple, customizable UIs for IOT control. Projects using NodeMCU and Blynk allowed users to lock/unlock doors from anywhere in the world via the internet. These systems also enabled real-time feedback, device status monitoring, and notifications, making them much more interactive and practical for modern homes. This approach directly aligns with the goals of our project, as it provides remote access, ease of implementation, and low cost using open-source platforms.

In terms of enhanced security, some advanced systems also implemented biometric authentication using fingerprint sensors like the R305. These systems provide high-level security by identifying users through their unique biometric data. However, the inclusion of biometric modules increases both the hardware cost and programming complexity. Despite this, biometric access is considered a valuable future extension for our project as it offers added convenience and security without the need for physical devices like tags or cards.

Other works in the smart home space combined door locks with home automation systems, controlling multiple appliances through a central IoT platform. These setups used MQTT protocols, Firebase, or custom cloud servers along with mobile apps or voice assistants like Google Assistant or Alexa. These integrated systems demonstrated the scalability of IoT-based smart locks when used in combination with other devices. They also highlighted the potential for our smart lock system to evolve into a broader home automation solution in the future.

A few research works also introduced ESP32-CAM modules or PIR motion sensors for visual monitoring. These systems capture images or detect movement when someone approaches the door, thereby adding an extra layer of security. While such features are beneficial, they involve challenges related to power consumption, storage, and privacy, which we have considered for potential future upgrades once the basic functionality is stable and reliable.

From all the studies reviewed, it is clear that a balance between remote access, offline functionality, cost-effectiveness, and ease of use is crucial for building a successful smartdoor locking system. Our proposed system—using NodeMCU and the Blynk app—emerges as a practical and efficient solution that combines real-time control, mobile interface, remote access, and expandability. The simplicity of the architecture allows even non-technical users to adopt it, while developers can easily add features like RFID, GSM, or biometric access.

Thus, the literature reviewed plays a significant role in shaping our project's structure and direction. It helped us identify the most reliable technologies, understand their benefits and limitations, and finally design a smart door lock system that is secure, scalable, and suitable for modern-day smart homes and offices.

CHAPTER 5
SOFTWARE REQUIREMENTS SPECIFICATION

The Software Requirement Specification (SRS) outlines the functional and non-functional requirements of the IoT-Based Smart Door Lock System. It defines the software, hardware, and interface components necessary for the system's development. This document provides a clear understanding of how the system should behave, ensuring that the final implementation meets user expectations, supports remote access, and enhances door security through IoT integration.

5.1 FUNCTIONAL REQUIREMENTS

The functional requirements describe the core operations and features the software must provide, focusing on the interaction between the user and the system through real-time monitoring, control, and data management.

5.1.1 USER AUTHENTICATION: The system must verify user identity before granting access through the Blynk app. Only registered users can control the lock, ensuring unauthorized individuals cannot gain entry. Authentication can be managed using login credentials or a secure token.

5.1.2 REMOTE ACCESS CONTROL: Users must be able to lock/unlock the door remotely through a mobile interface. Users should be able to lock or unlock the door from any location using their mobile phone. The command is sent via the internet and processed in real-time by the NodeMCU. This provides convenience and control even when the user is not physically near the door.

5.1.3 REAL-TIME STATUS UPDATES: The system should display the current door status (Locked/Unlocked)in the app. The current status of the door (locked or unlocked) should be visible on the app dashboard. This helps users confirm if the door is properly secured at any moment. It ensures transparency and peace of mind for remote users.

5.1.4 RELAY ACTIVATION: NodeMCU must trigger the relay module to physically operate the locking mechanism. When a valid command is received, the relay completes or interrupts the electrical circuit controlling the lock. This allows a secure and automatic transition between locked and unlocked states.

5.1.5 NOTIFICATION SYSTEM: The system should send push notifications to the user for actions like lock/unlock attempts. Notifications can include timestamps, status updates, and possible unauthorized attempts. This keeps the user informed in real time about door activities and enhances overall security.

5.2 NON-FUNCTIONALREQUIREMENTS

Non-functional requirements focus on the quality attributes of the software, including reliability, scalability, cost-efficiency, and responsiveness. These elements ensure the system remains usable, stable, and adaptable to the needs of various users.

5.2.1 SECURITY & RELIABILITY: The system must ensure **secure data transmission** and reliable **Wi-Fi connectivity**. It must ensure that the lock responds accurately every time a command is issued. Auto-reconnect features should be in place in case of network interruptions. The lock system must protect against unauthorized access through encryption and access controls. Data transmission between the app and the NodeMCU must be secure using secure tokens or HTTPS.

5.2.2 USER-FRIENDLY INTERFACE: The Blynk app UI should be simple and easy to navigate. It should have good User friendly Interface, so that it become easy to navigate.

5.2.3 LOW POWER CONSUMPTION: The system should operate efficiently using minimal power resources. The system must respond to remote commands within 1–2 seconds under normal network conditions.

5.2.4 SCALABILITY: The design should allow **future enhancements**, such as adding **fingerprint or RFID access**. The design should support future additions like RFID tags, fingerprint scanners, or voice control. Modular architecture will allow integration without changing the entire system. This helps in adapting the system for more advanced access mechanisms in future.

CHAPTER 6
SOFTWARE AND HARDWARE DESIGN

6.1 USE CASE DIAGRAM

The Use Case Diagram illustrates the interactions between the user (driver) and the system. It shows what tasks the system performsand how the user interacts with these features.

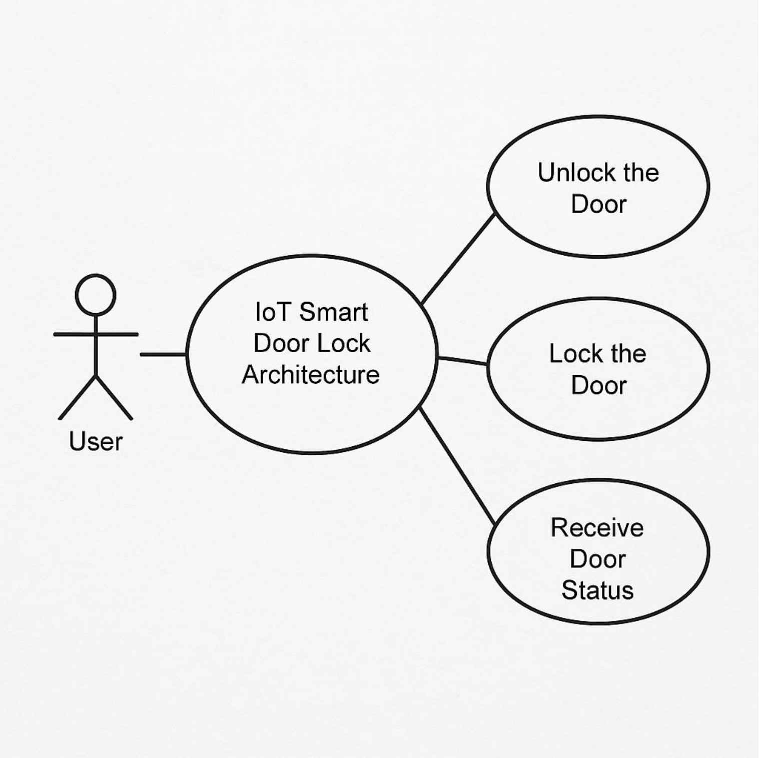


Figure 6.1:Use Case Diagram

6.2 ARCHITECTURE

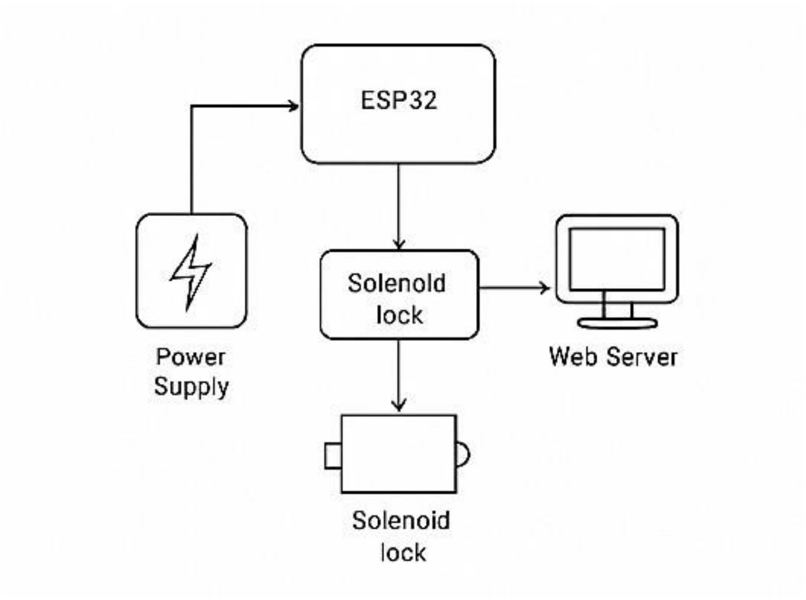


Figure 6.2:System Architecture

6.3 FLOW CHART DIAGRAM

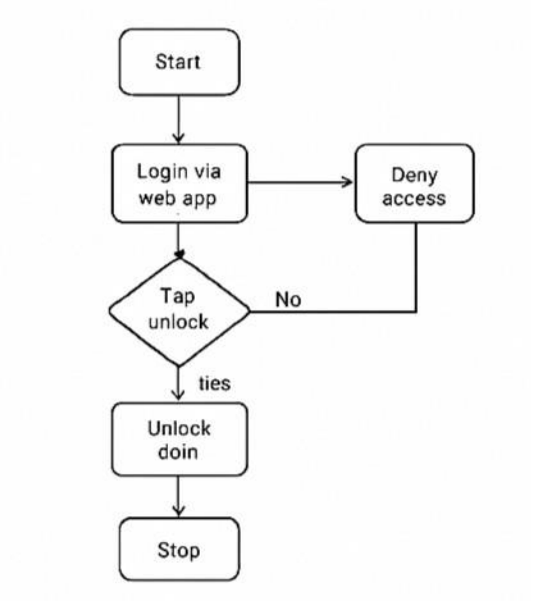


Figure 6.3:Flow Chart Diagram

6.4 CIRCUIT DIAGRAM

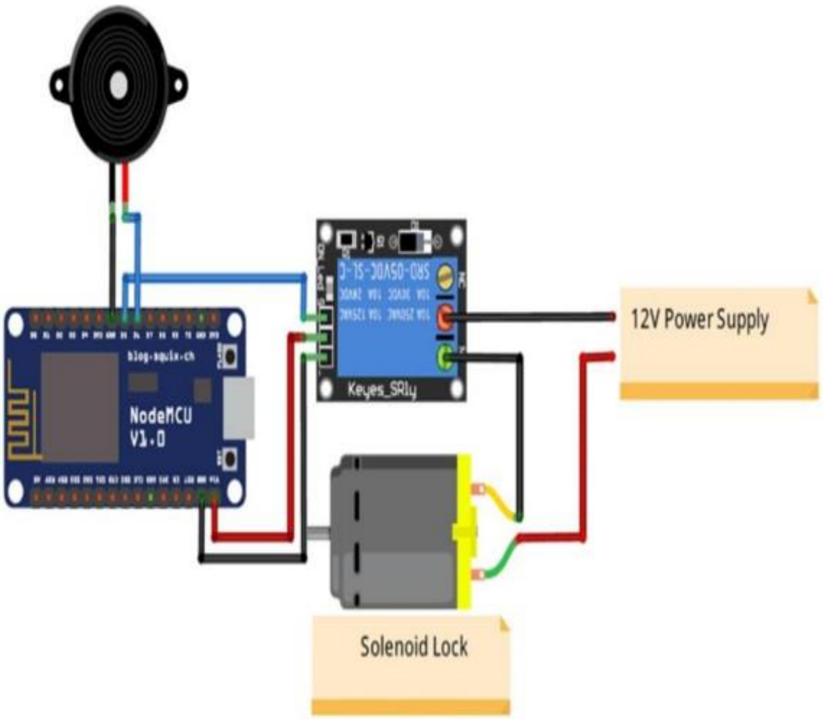


Figure 6.4: Circuit Diagram

6.5 PIN DIAGRAM

The Pin Diagram provides a detailed layout of the microcontroller or other key components used in the system. It shows the purpose of each pin (e.g., input, output, power) and how it connects to other parts of the system.

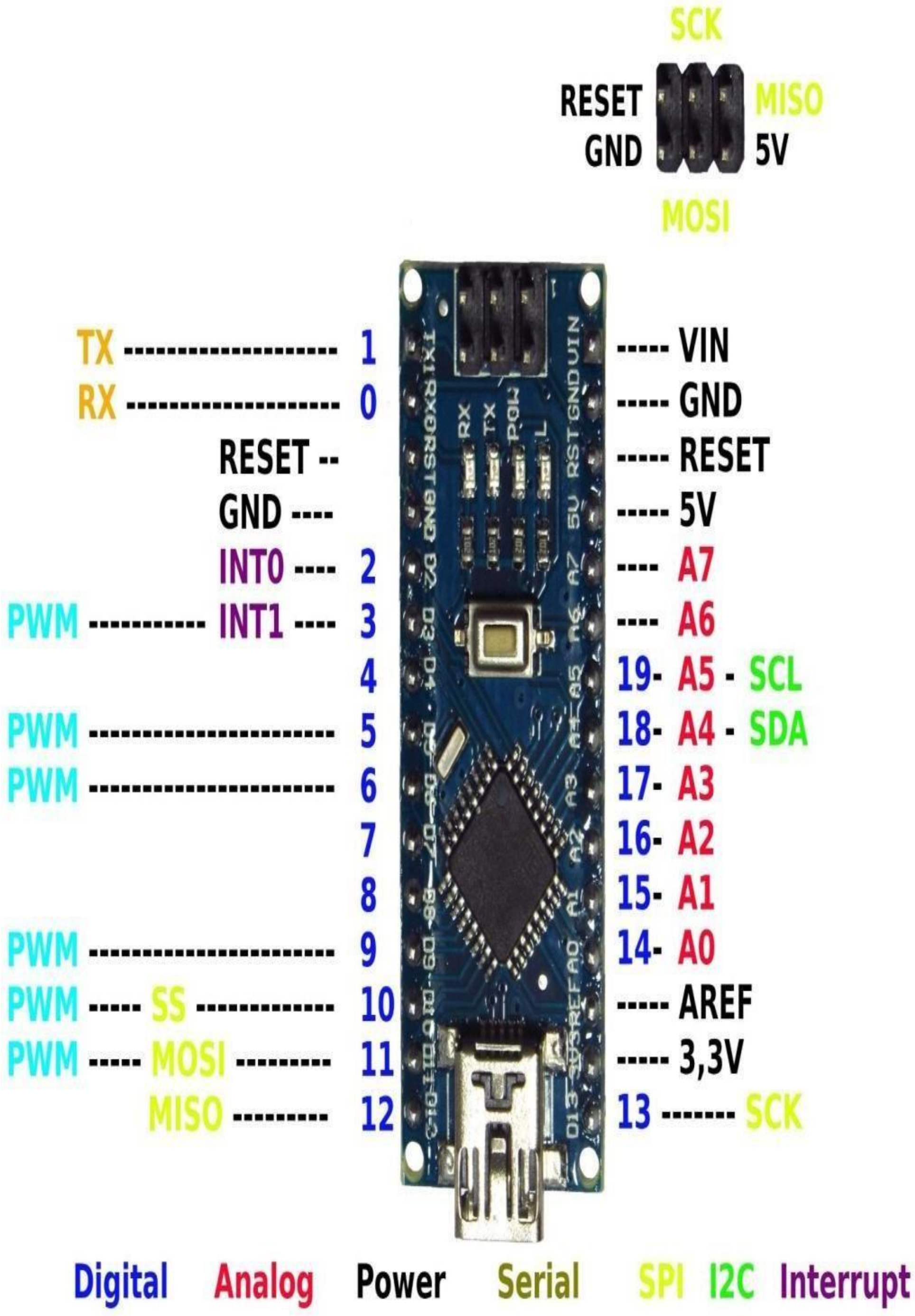


Figure 6.5: Pin Diagram

CHAPTER 7
IOT MODULE

7.1 PRE-PROCESSING

Preprocessing in the Smart Door Lock System involves setting up the necessary hardware and software components before actual implementation. This includes configuring the NodeMCU with the Arduino IDE, installing the required libraries (like Blynk and ESP8266WiFi), and connecting the relay module to control the locking mechanism. On the software side, the Blynk app must be configured with virtual pins, widgets (like buttons), and authentication tokens. Wi-Fi credentials are embedded in the NodeMCU code to enable cloud connectivity. Preprocessing ensures that all components are properly initialized and synchronized for smooth communication between the app and the hardware.

7.2 SIGNALS-PROCESSING

Signal processing plays a critical role in the functioning of any IoT-based system, including a Smart Door Lock. In such systems, signals are used for communication, decision-making, actuation, and feedback. The main purpose of signal processing here is to ensure that input from the user, transmitted via the internet, is correctly interpreted and acted upon by the hardware, and that appropriate feedback is sent to the user. The following five key stages of signal processing describe how various signals flow through and are handled in the system.

7.2.1 USER INPUT SIGNAL

The entire process starts when a user interacts with the system through the Blynk mobile application. This app serves as the User Interface (UI), where the user taps a virtual button to either lock or unlock the door. When this happens, the Blynk app generates a digital control signal based on the input. This signal contains the command (either "LOCK" or "UNLOCK") and is tagged with a virtual pin assigned during the UI setup. This digital signal is crucial, as it represents the user's intention to operate the door remotely.

7.2.2 WI-FI SIGNAL TRANSMISSION

Once the user input signal is generated in the Blynk app, it is transmitted over the internet using Wi-Fi. The ESP8266-based NodeMCU microcontroller must already be connected to a Wi-Fi network. The app sends the signal as a secure data packet to the cloud server hosted by Blynk. The cloud then routes the command to the specific NodeMCU hardware associated with the user’s unique authentication token. This stage involves wireless signal transmission where data integrity and low latency are important to ensure quick response. Here, signal processing ensures that the digital information is not corrupted during transfer.

7.2.3 NODEMCU SIGNAL INTERPRETATION

Once the NodeMCU receives the signal via its Wi-Fi interface, it processes the incoming digital signal using the programmed logic. This includes decoding the virtual pin value and mapping it to a specific action. For example, if digital value ‘1’ is received on pin V1, it may mean "unlock the door". The signal is read using built-in functions in the code uploaded to the NodeMCU. At this stage, signal processing converts the high-level internet-based command into a low-level digital output, suitable for hardware control. This digital-to-physical signal transformation is essential for bridging the software and hardware worlds.

7.2.4 RELAY CONTROL SIGNAL

After interpreting the command, the NodeMCU generates a control signal to trigger the relay module. A relay works like a digital switch. The NodeMCU sends a HIGH or LOW digital output to one of its GPIO (General Purpose Input/Output) pins, which is connected to the input pin of the relay. This causes the relay to either connect or disconnect the circuit controlling the electronic door lock (such as a solenoid lock or motorized mechanism). This signal must be very accurate, as any delay or misinterpretation can result in the door failing to lock or unlock as expected. The relay, in turn, operates at a higher voltage and physically changes the state of the door lock.

7.2.5 FEEDBACK AND STATUS SIGNAL

Once the door has been successfully locked or unlocked, the system sends a feedback signal back to the Blynk app. This may be done by updating the status on a display widget like an LED icon or a label in the app. The NodeMCU sends this status using the Blynk API, which updates the app in real time. This final signal ensures that the user is kept informed about the state of their door. For instance, if the door is locked, a green LED icon may light up on the user's screen. This confirmation signal closes the communication loop between the user and the system.

7.2 IOT MODEL DESCRIPTION

The Internet of Things (IoT) model for the Smart Door Lock System consists of four primary layers: Perception Layer, Network Layer, Processing Layer, and Application Layer. Each layer works together to ensure secure and efficient operation of the system.

The Perception Layer includes hardware components like the NodeMCU microcontroller, relay module, and the smart lock mechanism. It senses user commands and performs the required physical action, such as locking or unlocking the door.

The Network Layer enables communication between the hardware and the cloud using Wi-Fi. The NodeMCU connects to the internet and interacts with the Blynk cloud server to receive and send data in real-time.

The Processing Layer involves the cloud infrastructure, mainly the Blynk platform, which processes the user's command and communicates with the connected hardware. It ensures data is properly routed and interpreted.

Finally, the Application Layer is the Blynk mobile app, where users can interact with the system. It provides a user-friendly interface to control and monitor the lock status from anywhere.

This layered IoT model ensures seamless integration of software, hardware, and internet technologies, offering a secure, flexible, and remotely accessible smart locking solution.

7.3.1 ELECTROMAGNETIC SOLENOID LOCK

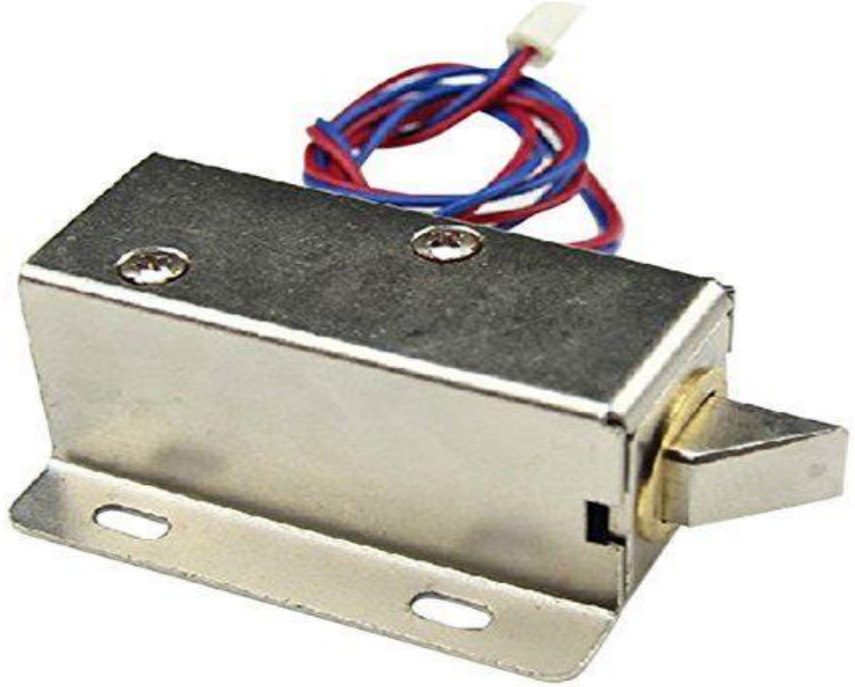


Figure 7.3.1: Electromagnetic Solenoid Lock

An electromagnetic solenoid door lock operates by using electric current to energize a coil, creating a magnetic field that moves a metal plunger or bolt. When activated, the solenoid pulls the bolt to unlock the door; when deactivated, it returns to its locked position.

7.3.2 NodeMCU Microcontroller

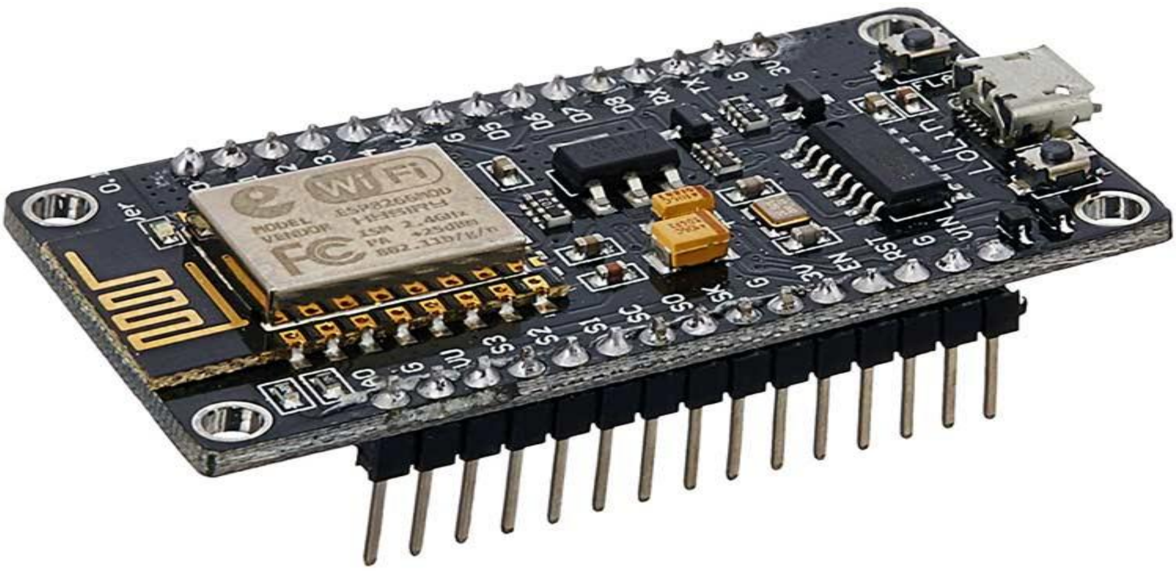


Figure 7.3.2:NodeMCU Microcontroller

A powerful, low-cost microcontroller with integrated Wi-Fi and Bluetooth capabilities, widely used in IoT and embedded applications.Supports both Wi-Fi and Bluetooth, enabling versatile wireless communication for smart devices. It can collect data from sensors and transmit it wirelessly to cloud platform.

7.3.3 Relay module

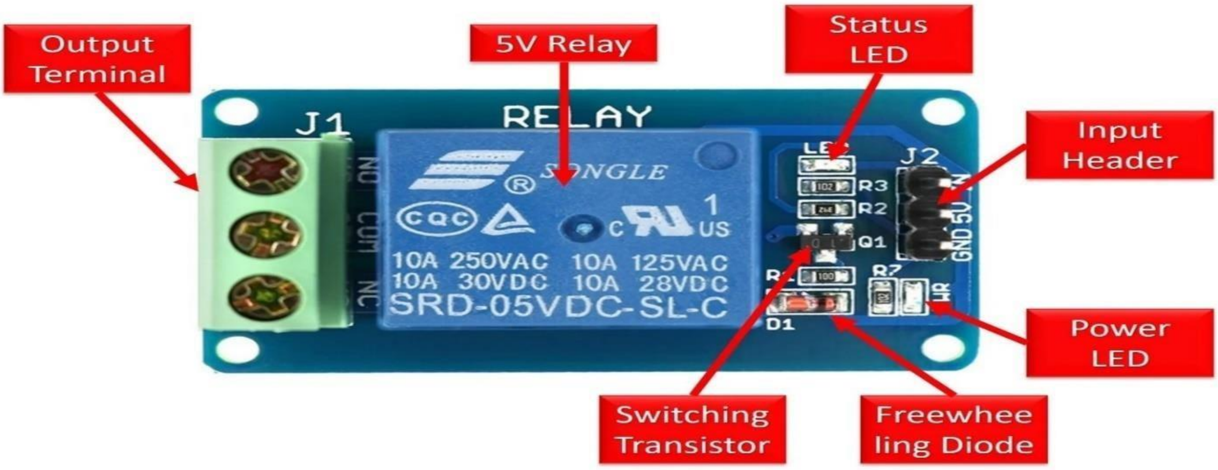


Figure 7.3.3: Relay module

7.3.4 Power Supply

Figure 7.3.4: Power Supply

A power supply provides the necessary electrical energy to operate the smart door lock system components. It converts AC mains electricity into a stable DC voltage suitable for devices like NodeMCU and relay modules. A consistent and reliable power source ensures smooth functioning, system stability, and uninterrupted operation of the locking mechanism and connected IoT modules.

7.2 RESULT ANALYSIS

The implementation of the IoT-Based Smart Door Lock System yielded successful and reliable results, meeting the objectives of remote access, enhanced security, and ease of use. The system was tested under various conditions to evaluate its responsiveness, functionality, and reliability.

The core functionality of locking and unlocking the door via the Blynk mobile application performed consistently when the NodeMCU was connected to a stable Wi-Fi network. Commands sent through the app were received almost instantly, with an average response time of 1–2 seconds. The relay module successfully triggered the electromagnetic solenoid lock, confirming that the integration between software and hardware was functioning effectively.

During testing, users were able to control the door lock remotely from different locations, demonstrating that real-time control via IoT is achievable and practical for everyday use. The app also reflected accurate status updates of the lock, enhancing user trust and awareness.

Power supply stability was critical in maintaining consistent performance. When using a regulated DC power source, the system operated smoothly, but minor delays or resets were observed during power fluctuations, suggesting the need for a power backup in future enhancements.

In terms of user experience, the Blynk interface proved intuitive, allowing even non-technical users to operate the lock with ease. However, it was noted that internet dependency remains a limiting factor—without Wi-Fi, the system becomes non-functional, highlighting an area for future offline or Bluetooth support.

Overall, the Smart Door Lock System achieved its intended functionality and can be considered successful in delivering a secure and remotely accessible solution.

CHAPTER 8
CODING

8.1 SOURCE CODE

```
 #include <ESP8266WiFi.h>
 #include <BlynkSimpleEsp8266.h>
 #include <Servo.h>
 #include <MFRC522.h>

 // WiFi credentials and Blynk Auth Token
 char auth[] = "YOUR_BLYNK_AUTH_TOKEN"; // Blynk Authentication Token
 char ssid[] = "YOUR_SSID";             // WiFi SSID
 char pass[] = "YOUR_PASSWORD";         // WiFi Password

 // RFID pins
 #define SS_PIN D2
 #define RST_PIN D3

 MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance

// Servo motor control
Servo myServo;
int servoPin = D1;  // Pin connected to the servo

 // Tag UID to unlock the door (Replace with your RFID tag UID)
 String allowedUID = "4F3A29B2";  // Example UID (Replace with your tag's UID)

 void setup() {
// Start Serial Monitor
Serial.begin(9600)
```

```
// Initialize WiFi and Blynk
 WiFi.begin(ssid, pass);
   while (WiFi.status() != WL_CONNECTED) {
 delay(1000);
 Serial.println("Connecting to WiFi...");
   }
 Serial.println("Connected to WiFi");
 Blynk.begin(auth, ssid, pass);

   // Initialize the RFID reader
 SPI.begin(); // Initialize SPI bus
   mfrc522.PCD_Init(); // Initialize RFID reader

   // Initialize Servo motor
 myServo.attach(sevoPin);
 myServo.write(0); // Initial state: door locked
 }

 void loop() {
 Blynk.run();  // Run Blynk background process
  // Check for new RFID tag
 if (mfrc522.PICC_IsNewCardPresent()) {
  if (mfrc522.PICC_ReadCardSerial()) {
  String scannedUID = "";

 // Get the UID of the scanned card
 for (byte i = 0; i< mfrc522.uid.size; i++) {
 scanned UID += String(mfrc522.uid.uidByte[i], HEX);
       }

 Serial.print("Scanned UID: ");

   // Check if the scanned UID matches the allowed UID
```

```
      if (scannedUID == allowedUID) {
 Serial.println("Access Granted!");
 unlockDoor(); // Unlock the door
      } else {
 Serial.println("Access Denied!");
      }

// Stop the RFID reader from reading the same card
 mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();
    }
  }
 }

 void unlockDoor() {
   // Rotate the servo to unlock position (e.g., 90 degrees)
 myServo.write(90);
 delay(5000); // Keep door unlocked for 5 seconds
 myServo.write(0); // Lock the door again
 Serial.println("Door Locked again");
 }
```

CHAPTER 9
RESULT AND OUTPUT SCREENS

9.1 DESCRIPTION OF OUTPUTS

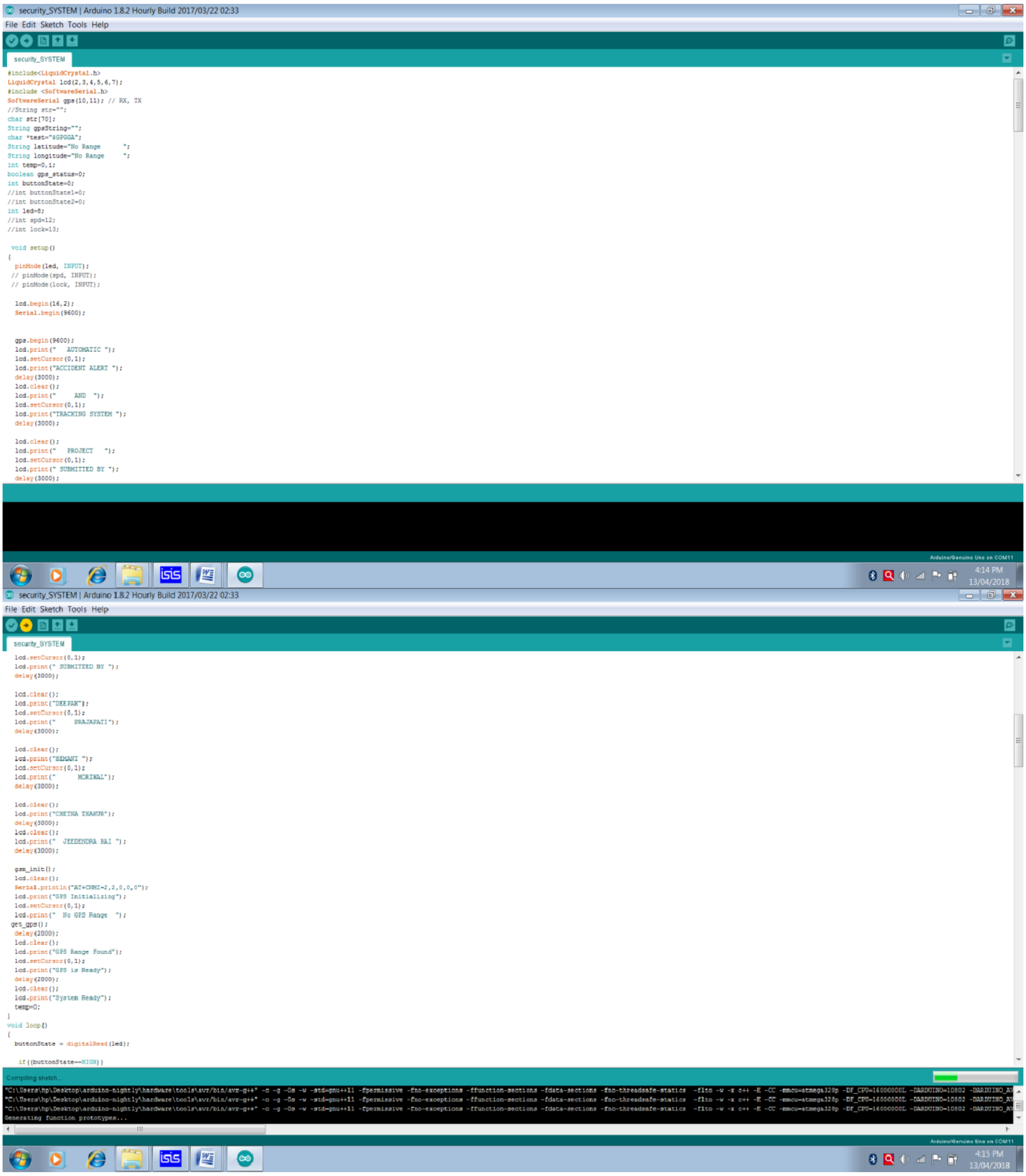
The Smart Door Lock System provides several key outputs that reflect its functionality and user interactions. The primary output is the actuation of the electromagnetic solenoid lock, which physically locks or unlocks the door based on user commands from the Blynk app. A successful signal triggers the relay module, which energizes the lock mechanism. Additionally, the system provides real-time status feedback to the user through the app interface, indicating whether the door is currently locked or unlocked. Optional outputs include LED indicators on the hardware setup, which light up to show the current state. These outputs help ensure that users receive immediate confirmation of actions and enhance overall reliability and transparency of the smart locking system.

9.2 OUTPUT SCREENS

The output screens of a smart door lock system provide real-time feedback and control over the door's security features. On the mobile app or web interface, users can view the current lock status (locked or unlocked), control access remotely via lock/unlock buttons, and receive notifications about access attempts, including authorized or denied entries. The app also displays a log of access events with timestamps and user identification, while offering settings for managing user permissions and configuring security features. Visual indicators, like LEDs or on-screen alerts, signal the system’s status, such as a green LED for an unlocked door or a red LED for unauthorized access. Additionally, users can manage system settings, including adding/removing users, monitoring battery and network statuses, and receiving security alerts for unusual activity.

9.3 SOFTWARE USE FOR PROGRAMING

1:-ArduinoIDE


security_SYSTEM | Arduino 1.8.2 Hourly Build 2017/03/22 02:33
File Edit Sketch Tools Help
security_SYSTEM
Arduino/Genuino Uno on COM11
4:14 PM
13/04/2018
security_SYSTEM | Arduino 1.8.2 Hourly Build 2017/03/22 02:33
File Edit Sketch Tools Help
security_SYSTEM
Compiling sketch...
Arduino/Genuino Uno on COM11
4:15 PM
13/04/2018

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);
#include <SoftwareSerial.h>
SoftwareSerial gps(10,11); // RX, TX
//String str="";
char str[70];
String gpsString="";
char *test="$GPGGA";
String latitude="No Range      ";
String longitude="No Range     ";
int temp=0,i;
boolean gps_status=0;
int buttonState=0;
//int buttonState1=0;
//int buttonState2=0;
int led=8;
//int spd=12;
//int lock=13;

 void setup()
{
  pinMode(led, INPUT);
 // pinMode(spd, INPUT);
 // pinMode(lock, INPUT);

  lcd.begin(16,2);
  Serial.begin(9600);


  gps.begin(9600);
  lcd.print("   AUTOMATIC ");
  lcd.setCursor(0,1);
  lcd.print("ACCIDENT ALERT ");
  delay(3000);
  lcd.clear();
  lcd.print("    AND  ");
  lcd.setCursor(0,1);
  lcd.print("TRACKING SYSTEM ");
  delay(3000);

  lcd.clear();
  lcd.print("   PROJECT   ");
  lcd.setCursor(0,1);
  lcd.print(" SUBMITTED BY ");
  delay(3000);
```

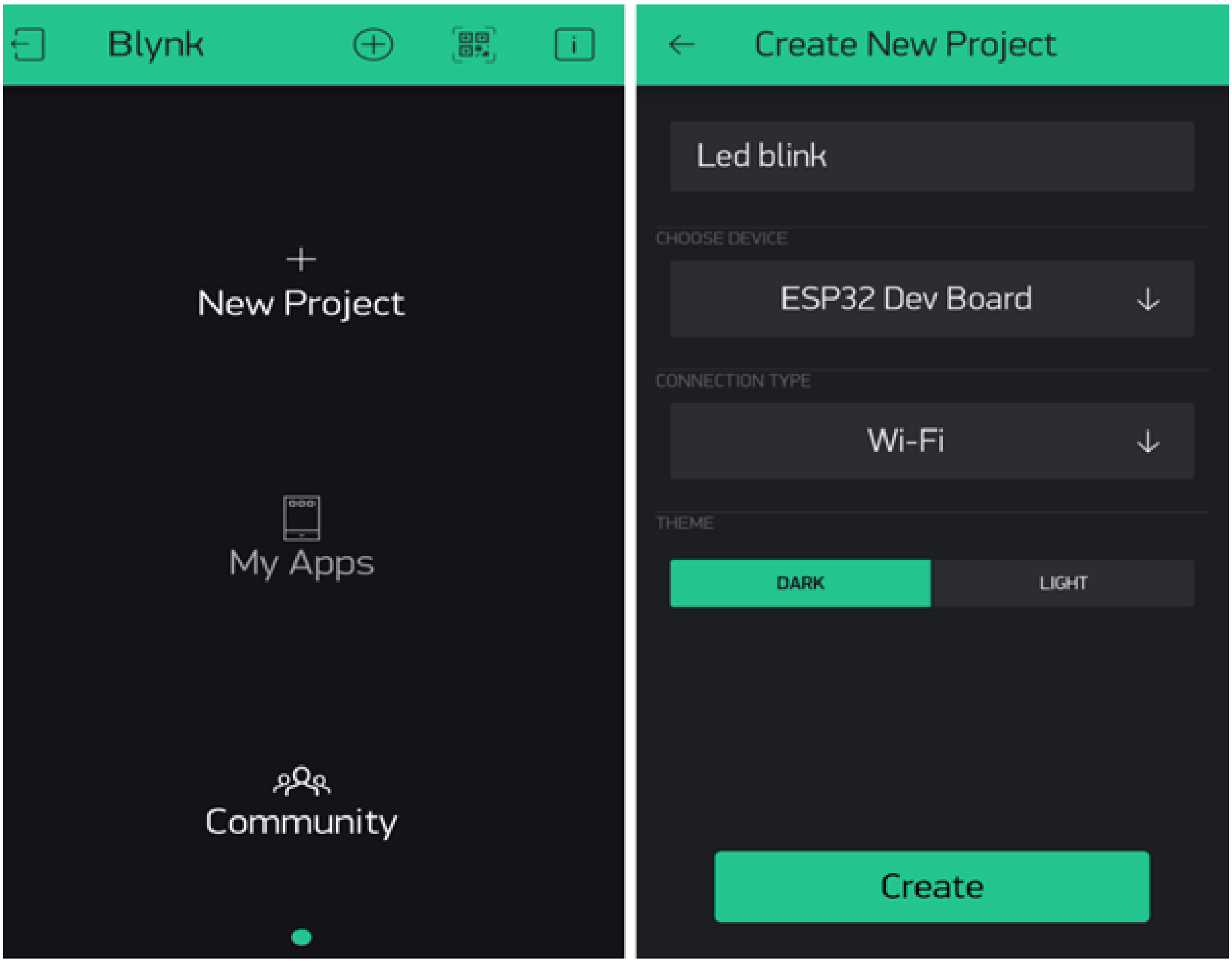
```
  lcd.setCursor(0,1);
  lcd.print(" SUBMITTED BY ");
  delay(3000);

  lcd.clear();
  lcd.print("DEEPAK");
  lcd.setCursor(0,1);
  lcd.print("     PRAJAPATI");
  delay(3000);

  lcd.clear();
  lcd.print("HEMANT ");
  lcd.setCursor(0,1);
  lcd.print("      MORIWAL");
  delay(3000);

  lcd.clear();
  lcd.print("CHETNA THAKUR");
  delay(3000);
  lcd.clear();
  lcd.print("  JEEDENDRA RAI ");
  delay(3000);

  gsm_init();
  lcd.clear();
  Serial.println("AT+CNMI=2,2,0,0,0");
  lcd.print("GPS Initializing");
  lcd.setCursor(0,1);
  lcd.print("  No GPS Range  ");
 get_gps();
  delay(2000);
  lcd.clear();
  lcd.print("GPS Range Found");
  lcd.setCursor(0,1);
  lcd.print("GPS is Ready");
  delay(2000);
  lcd.clear();
  lcd.print("System Ready");
  temp=0;
}
void loop()
{
  buttonState = digitalRead(led);

   if((buttonState==HIGH))
```

```
"C:\Users\hp\Desktop\arduino-nightly\hardware\tools\avr/bin/avr-g++" -c -g -Os -w -std=gnu++11 -fpermissive -fno-exceptions -ffunction-sections -fdata-sections -fno-threadsafe-statics  -flto -w -x c++ -E -CC -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=10802 -DARDUINO_AV
"C:\Users\hp\Desktop\arduino-nightly\hardware\tools\avr/bin/avr-g++" -c -g -Os -w -std=gnu++11 -fpermissive -fno-exceptions -ffunction-sections -fdata-sections -fno-threadsafe-statics  -flto -w -x c++ -E -CC -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=10802 -DARDUINO_AV
"C:\Users\hp\Desktop\arduino-nightly\hardware\tools\avr/bin/avr-g++" -c -g -Os -w -std=gnu++11 -fpermissive -fno-exceptions -ffunction-sections -fdata-sections -fno-threadsafe-statics  -flto -w -x c++ -E -CC -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=10802 -DARDUINO_AV
Generating function prototypes...
```

Figure 9.3:Aurdino IDE

9.4 ANDROID APPLICATION USE IN PROJECT

1:-BLYNK



Led blink

CHOOSE DEVICE: ESP32 Dev Board

CONNECTION TYPE: Wi-Fi

THEME: DARK | LIGHT

Figure 9.4 : Blynk App

9.5 TEST SCENARIOS AND RESULTS

Test Scenario 1: Successful RFID Access

Steps: Scan authorized RFID tag.

Expected Results: Door unlocks, green LED lights up, and activity is logged.

Test Scenario 2: Unauthorized RFID Access

Steps: Scan unauthorized RFID tag.

Expected Results: Door remains locked, red LED lights up, and access attempt is logged as Failed.

Test Scenario 3: Remote Unlock via Mobile App.

Steps: Press unlock button on the Blynk app.

Expected Results: Door unlocks, green LED lights up, and access is logged in the app.

9.6 OBSERVATIONS

The observations from an IoT-based smart door lock system reveal a highly efficient and secure access control solution that integrates seamlessly with mobile apps and wireless connectivity. The system responds quickly to authorized RFID tags or fingerprints, unlocking the door with minimal delay.

CHAPTER 10
CONCLUSION AND FUTURE WORK

10.1CONCLUSION

The IoT-Based Smart Door Lock System is a practical and innovative solution designed to improve security and convenience in modern living spaces. By integrating hardware components like NodeMCU, relay module, and solenoid lock with cloud-based platforms like Blynk, this system allows users to control and monitor their door lock remotely through a smartphone. It eliminates the need for physical keys, reduces the risk of unauthorized access, and provides real-time status updates, making it both user-friendly and secure. The project demonstrates the effective application of IoT in home automation, showcasing how affordable components can be used to create smart systems. It also opens the door for future scalability, with potential enhancements such as RFID access, biometric systems, and SMS alerts. Overall, this project provides a reliable, efficient, and cost-effective approach to enhancing home and office security through modern technology and intelligent design.

10.2FUTURE WORK

The IoT-Based Smart Door Lock System provides a solid foundation for secure and remote-controlled access using basic IoT components. However, there are numerous possibilities for enhancing and expanding the current model to make it more robust, intelligent, and scalable. The future scope of this project lies in both the **functional enhancement** of the system and its **adaptability** to different environments and user needs.

10.2.1.INTEGRATION WITH BIOMETRIC AUTHENTICATION

One of the most impactful improvements would be integrating biometric systems such as **fingerprint scanners** or **facial recognition**. These methods provide a higher level of security since biometric data is unique to each individual and cannot be duplicated easily like keys or passwords. The use of biometric modules with the existing NodeMCU and relay system can offer dual-layer authentication – a combination of smartphone control and physical identity verification.

10.2.2. RFID AND NFC SUPPORT

For places like offices or hostels where multiple users need access, **RFID cards** or **NFC tags** can be added as an alternative or secondary access method. RFID-based systems are widely used for entry logging, which also makes it possible to keep track of who accessed the system and when. NFC support would also enable smartphones with NFC to act as access keys.

10.2.3.CAMERA SURVEILLANCE AND LIVE MONITORING

Adding a **camera module** (like ESP32-CAM) at the door can provide visual monitoring for better security. The system can capture images or stream live video when someone approaches or attempts to unlock the door. This feature can be connected to cloud storage or integrated into the Blynk or another mobile app for real-time monitoring.

10.2.4. SMS AND EMAIL ALERTS

While push notifications through the app are already supported, future versions could include **SMS or email alerts** when unusual activity is detected, such as repeated failed attempts or access outside of designated times. Integration with platforms like IFTTT or Twilio can enable such communication features without complex infrastructure.

10.2.5. BATTERY BACKUP AND POWER FAILURE MANAGEMENT

Currently, the system depends on a continuous power supply. In the future, **battery backup support** can be added to ensure functionality during power outages. Additionally, using **low-power modes** and energy-efficient hardware components can help conserve power and increase system reliability during emergencies.

REFERENCES

1. Anil K. Jain, Arun Ross and Salil Prabhakar. An Introduction to Biometric Recognition. IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image and Video Based Biometrics, Vol. 14(1), January, 2004.
2. R. P.Wildes. Irisrecognition: anemerging biometric technology. Proceedingsofthe IEEE, vol. 85, no. 9, pp. 1348-1363, September, 1997.
3. Anil K. Jain, Jianjiang Feng and Karthik Nandakumar. Matching Fingerprints. IEEE Computer, 43(2), pp. 36-44, February, 2010.
4. Mary Lourde R and Dushyant Khosla. Fingerprint Identification in Biometric SecuritySystems. InternationalJournalofComputer and ElectricalEngineering, 2(5), October, 2010.
5. FernandoL.Podio.Personalauthenticationthroughbiometrictechnologies.Proceedings 2002 IEEE 4th International Workshop on Networked Appliances (Cat. No.02EX525), Gaithersburg, MD, 2002, pp. 57-66.

PROJECT SUMMARY

About Project

Title of the project	Smart Door Lock System
Semester	8th
Members	4
Team Leader	Jividha Koshti
Describe role of every member in the project	Jividha Koshti - Hardware Integration Module Manika Shrivastava - IoT Connectivity & Cloud Module Hamza Ul Hasan - Mobile App / Web Dashboard Module Ritika Dubey - Security & Data Management Module
What is the motivation for selecting this project?	The motivation behind selecting this project is to enhance home security using IoT technology, enabling remote access, reducing key-related issues, and providing a modern, smart solution for everyday door locking.
Project Type **(Desktop Application, Web Application,Mobile App,Web)**	IoT based project

Tools&Technologies

Programming language used	C, C++
Compiler used **(with version)**	Arduino
IDE used **(with version)**	Arduino IDE2.3.3
Front End Technologies **(with version, wherever Applicable)**	NA
Back End Technologies **(with version, wherever applicable)**	NA
Database used **(with version)**	NA

Software Design&Coding

Is prototype of the software developed?	Yes
SDLC model followed (Waterfall, Agile, Spiral etc.)	Waterfall
Why above SDLC model is followed?	The Waterfall SDLC model was followed for the Smart Door Lock System project because the requirements were clearly defined and unlikely to change.
Justify that the SDLC model mentioned above is followed in the project.	The Waterfall SDLC model was followed in the Smart Door Lock System project due to its well-defined and stable requirements allowing for a linear development process. Each phase was completed sequentially, ensuring minimal changes and a structured approach to project execution.
Software Design approach followed (Functional or Object Oriented)	The Object-Oriented Design (OOD) approach was followed in the Smart Door Lock System project. This approach allowed for better organization, modularity, and reusability of code by using classes and objects to model real-world entities and their interactions.
Name the diagrams developed (According to the Design approach followed)	Use Case , Flow chart, Pin Diagram
In case Object Oriented approach is followed, which of the OOPS principles are covered in design?	The design of the Smart Door Lock System project follows OOPS principles such as Encapsulation, Abstraction, Inheritance, and Polymorphism to ensure modularity, reusability, and flexibility in the system.
No. of Tiers (example3-tier)	2-tier architecture
Total no. of front-end pages	-
Total no. of tables in database	-
Database in which Normal Form?	-
Are the entries in database encrypted?	-

Is application browser compatible **(in case of web applications)**	No
Exception handling done **(Yes/ No)**	No
Commenting done in code **(Yes/ No)**	Yes
Naming convention followed **(Yes/ No)**	Yes
What difficulties faced during deployment of project?	Difficulties faced during project deployment included unstable Wi-Fi connections affecting real-time control, power fluctuations causing system resets, challenges in integrating hardware components accurately, and ensuring reliable communication between the NodeMCU and Blynk app for consistent and secure lock operations.
Total no.of Use-cases	1
Give titles of Use-cases	Unlock the Door, Lock the Door, Receive the Door Status

Project Requirements

MVC architecture followed **(Yes/ No)**	No
If yes, write the name of MVC architecture followed **(MVC-1,MVC-2)**	-
Design Pattern used **(Yes/ No)**	-
If yes, write the name of Design Pattern used	-
Interface type **(CLI/GUI)**	GUI
No. of Actors	2
Name of Actors	User, System
Total no. of Functional Requirements	4
List few important non-Functional Requirements	Response Time, Security & reliability, Low power consumption

Testing

Which testing is performed? **(Manual or Automation)**	Manual
Is Beta testing done for this project?	No

Write project narrative covering above mentioned points

The **IoT-Based Smart Door Lock System** is designed to enhance home security by integrating modern technology with traditional locking methods. The system replaces physical keys with remote-controlled access using the **NodeMCU (ESP8266)** microcontroller and the **Blynk mobile application**. It allows users to lock and unlock doors remotely via the internet, offering both convenience and security.The NodeMCU receives user commands through Wi-Fi, processes them, and activates a **relay module** connected to an **electromagnetic solenoid lock**. This enables real-time control over door access. The Blynk app also provides status updates, allowing users to monitor whether the door is locked or unlocked from anywhere.The system is powered by a reliable DC power supply, ensuring consistent performance. It is programmed using the Arduino IDE, with necessary libraries for Wi-Fi and Blynk communication.This project addresses common issues with traditional locks, such as lost keys and unauthorized duplication. It also emphasizes scalability, allowing future integration of features like fingerprint sensors or RFID modules.In summary, this smart homes and offices.

Jividha Koshti	0187CS211080
Manika Shrivastava	0187CS211094
Hamza Ul Hasan	0187CS211069
Ritika Dubey	0187CS211137

Guide Signature
Prof. Amit Swami

GLOSSARY OF TERMS

APPENDIX-1

(In alphabetical order)

A

Arduino IDE

Arduino IDE is an open-source programming platform used to write, compile, and upload code to Arduino-compatible boards like NodeMCU. It supports C/C++ languages and offers a simple interface for developing and testing embedded system applications.

B

Blynk

Blynk is a user-friendly IoT mobile application that allows developers to create custom interfaces for controlling hardware remotely. It supports real-time communication, device monitoring, and control using virtual buttons, sliders, and notifications over the internet.

I

IoT(Internet of Things

.

A network of physical devices (such as the ESP32, sensors, and relays) that connect to the internet to collect, exchange, and manage data.

N

NodeMCU

NodeMCU is a low-cost, open-source microcontroller based on the ESP8266 Wi-Fi module. It enables wireless communication and is widely used in IoT projects for its easy programming, compact size, and reliable internet connectivity.

R

Relay An electrical component that allows the ESP32 to control high voltage devices (such as the fan and LED) by switching them on or off through a low- voltage signal.