IT&BUSINESS COLLEGE

Department: COMPUTER SCIENCE
Project Title: SMART HOME SECURITY SYSTEM
Group Name: 0004

Team Members:
Toktogonov Daniel (ID: 238715027)
Akbaralieva Ademi (ID: 238715013)
Aidarova Raiana (ID: 238715034)
Asanova Nurperi (ID: 238715035)
Aibyn Nuramir (ID: 238715031)
Instructor: Nurbekov Mirlan
Group: SCA-23C

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. ACKNOWLEDGMENTS

We would like to express our gratitude to our teacher Mirlan Nurbekov for his guidance and support throughout the project we have been working on.

We are grateful to ALA-TOO IT & BUSINESS COLLEGE for providing the necessary funds and resources for this project.

We thank the open source development community, in particular the Arduino and Flutter communities, whose documentation and forums helped us overcome technical difficulties.

We thank every member of the team for their excellent work and teamwork. This project has been a rewarding learning experience for us, which has expanded our technical skills in developing embedded systems and mobile applications.

2. ABSTRACT

This project presents a Bluetooth-controlled smart home security system designed to detect unauthorized movement and alert users via a mobile application. The system addresses the need for affordable, customizable home security solutions by integrating ultrasonic motion sensors with Arduino Uno microcontroller and HC-06 Bluetooth module. When motion is detected, the system triggers audible alarms through buzzers, sends instant notifications to a Flutter-based mobile application, and allows authorized users to remotely control LED indicators, door locks via servo motors, and alarm systems based on their assigned permission levels. The final prototype successfully detects motion within sensor range, responds within acceptable latency, and provides role-based access control for four user types (User A, User B, User C, and Admin), demonstrating effective implementation of IoT security principles for residential applications.

3. INTRODUCTION & BACKGROUND

Home automation technology has become increasingly accessible for residential security applications. Smart home systems now provide real-time monitoring and remote control capabilities that were previously only available through expensive professional installations. This project focuses on home security, where automation technology can directly protect personal safety and property.

This security system is designed for students in dormitories, small apartments, and individuals who need simple motion detection without complex setup. The system detects unauthorized movement and sends alerts to users' mobile devices, allowing them to respond immediately regardless of their location within Bluetooth range.

Commercial systems like Ring Alarm and SimpliSafe offer comprehensive security features including cameras, professional monitoring, and cloud connectivity. However, these solutions require monthly subscriptions, stable internet connections, and often involve complicated installation. Arduino-based security projects exist but typically lack user-friendly mobile interfaces and permission management systems.

Our project provides a simpler alternative that works entirely through Bluetooth communication, eliminating internet dependency and subscription costs. The key difference is the implementation of role-based access control with four user types, each having specific permissions to control different components. This makes the system both practical for real security needs and valuable for learning embedded systems programming and mobile app development.

4. PROBLEM STATEMENT & OBJECTIVES

Problem Statement:

Many people living in dormitories, small apartments, or rented rooms cannot afford expensive professional security systems that require monthly subscriptions and stable internet connectivity. Existing DIY security solutions often lack user-friendly mobile interfaces and do not provide flexible permission management for multiple users. Students and individuals need a simple, cost-effective way to monitor their rooms and receive immediate alerts when unauthorized movement is detected. Current Arduino-based projects typically control all components through a single user interface without considering different access levels for family members or roommates. This project solves these problems by creating an affordable Bluetooth-controlled security system with role-based access control that works independently of internet connectivity.

Project Objectives:

1. Design a Bluetooth-controlled security system using Arduino Uno, HC-06 module, ultrasonic sensors, LEDs, buzzers, and servo motors for door locking.
2. Implement a Flutter mobile application that allows users to receive motion detection alerts and control system components based on their assigned permissions.
3. Develop a role-based access control system with four user types: User A (full control), User B (sensors only), User C (buzzers and servos), and Admin (permission management).
4. Test the system's detection accuracy, Bluetooth communication range, response time between motion detection and alert delivery, and reliability of component control.
5. Document the development process through multiple prototype versions showing incremental improvements from basic LED control to a complete multi-user security system.

5. SYSTEM REQUIREMENTS & CONSTRAINTS

5.1 Functional Requirements

The system must detect motion using three ultrasonic sensors and trigger alarm notifications immediately. When motion is detected, the system must send an alert to the mobile application and activate buzzers automatically. The mobile application must allow User A to control all components including 4 LEDs, 3 buzzers, 2 servo motors, and 3 sensors. User B must only have access to control the sensors through the app. User C must be able to control buzzers and servo motors for door locking. The Admin user must be able to grant or revoke permissions for any component and have the ability to shut down the entire system. All users must receive motion detection alerts regardless of their permission level. The system must maintain Bluetooth connection with the mobile device and respond to commands within acceptable time limits.

5.2 Non-Functional Requirements

The Bluetooth HC-06 module operates within a range of approximately 10 meters in open space, limiting the system's control distance. Response latency between sending a command from the mobile app and component activation should not exceed 2 seconds under normal conditions. The Arduino Uno operates at 5V with current consumption varying based on active components, requiring a stable power supply capable of providing at least 2A for simultaneous operation of all actuators. The total project cost must remain under $50 to maintain affordability as a student project. The system should operate continuously for at least 8 hours when powered by an external battery pack.

5.3 Mechanical Constraints

The prototype must fit within a compact enclosure not exceeding 20cm x 15cm x 10cm for easy placement in a room. Total system weight should remain under 500 grams to allow mounting on walls or furniture without additional support. Servo motors must provide sufficient torque to operate door lock mechanisms while maintaining compact size. Ultrasonic sensors require unobstructed line of sight and must be positioned to cover key entry points without creating blind spots.

5.4 Safety Requirements

All components must operate within safe voltage levels, with the main system running on 5V DC and servo motors not exceeding 6V. Maximum current draw for individual components must be limited to prevent overheating, with buzzers limited to 30mA and servo motors to 500mA each. Moving servo parts must be enclosed or positioned to prevent accidental contact during operation. The system must include overcurrent protection to prevent damage from short circuits. Buzzer

volume should remain below 85 dB to avoid hearing damage during extended alarm activation.

6. SYSTEM DESIGN

6.1 System Architecture

The system follows a hierarchical communication structure:

Mobile Application (Flutter) → Bluetooth (HC-06) → Microcontroller (Arduino Uno) → Actuators & Sensors

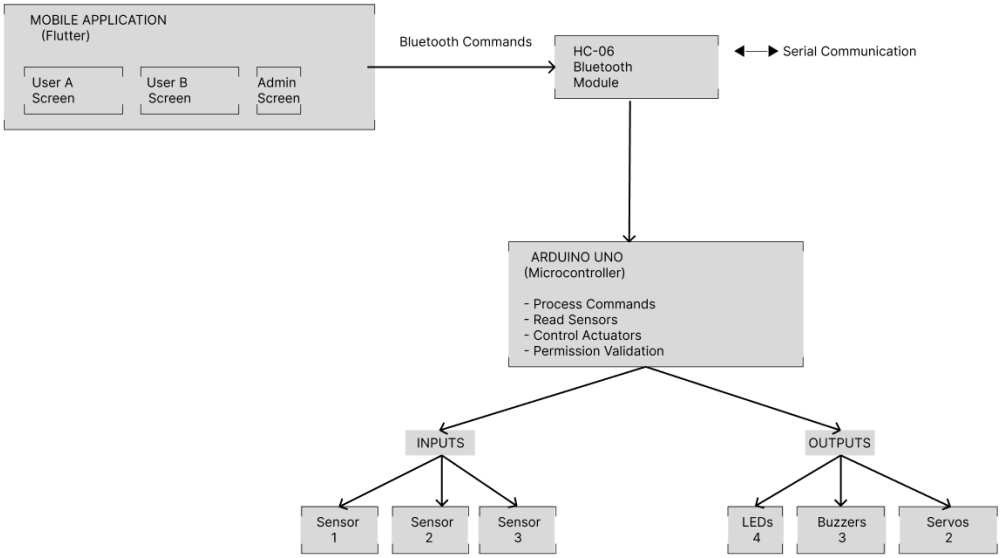

Figure 1:System Architecture Diagram

Description: Block diagram showing data flow from mobile app through Bluetooth module to Arduino, then to sensors and actuators with bidirectional communication arrows.

Table 1:**User Permission Matrix**

User	LEDs	Buzzers	Servos	Sensors	Admin Functions
User A	✓	✓	✓	✓	✗
User B	✗	✗	✗	✓	✗
User C	✗	✓	✓	✗	✗
Admin	✓	✓	✓	✓	✓

System Operation Flow:

1. **Initialization:** Arduino configures pins, initializes Bluetooth, sets component states
2. **Monitoring Loop:** Continuously reads sensors, checks for Bluetooth commands

3. **Motion Detection:** When sensor detects motion → Activate alarms → Send notification
4. **Command Processing:** Receive Bluetooth command → Validate permission → Execute action
5. **Response:** Send confirmation/error message back to mobile app

6.2 Hardware Design

Main Hardware Components:

Table 2:Component Specifications

Component	Model	Quantity	Voltage	Function
Microcontroller	Arduino Uno	1	5V	Central processing
Bluetooth Module	HC-06	1	3.3-5V	Wireless communication
Motion Sensor	HC-SR04 Ultrasonic	3	5V	Distance measurement
LED	Standard 5mm	4	5V	Visual indicators
Buzzer	Active Buzzer	3	5V	Audible alerts
Servo Motor	SG90	2	4.8-6V	Door locking
Resistor	220Ω	4	-	LED current limiting
Power Supply	USB/Wall Adapter	1	5V	System power

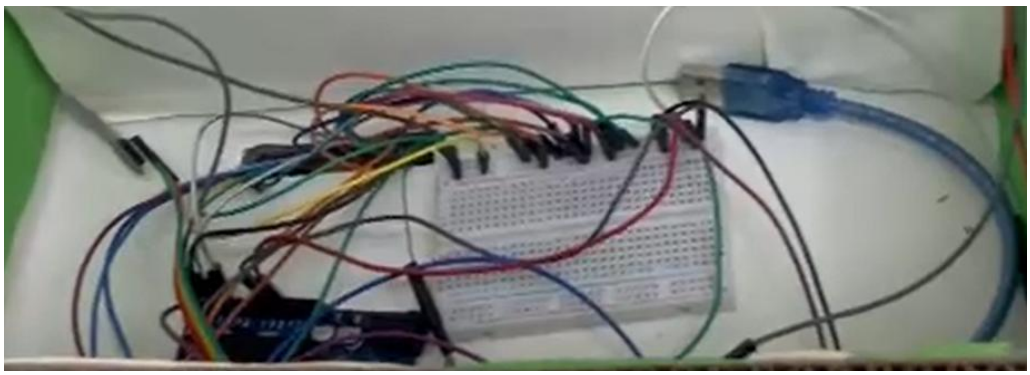

Figure 2:Hardware Component Layout

Description: Photo of breadboard showing all components connected with labeled wires and organized layout.

Pin Assignments:

- Digital Pin 11 (RX): HC-06 TX
- Digital Pin 10 (TX): HC-06 RX
- Digital Pins 2-4: Ultrasonic Trig pins
- Digital Pins 5-7: Ultrasonic Echo pins
- Digital Pins 8-9,12-13: LED outputs
- Digital Pins A0-A3: Buzzer outputs

- Digital Pins A4-A5: Servo control

Power Distribution:

- Arduino powered via USB (5V)
- Servos powered from external 5V supply with common ground
- All logic components share common ground

6.3 Software Design

Arduino Firmware Architecture:

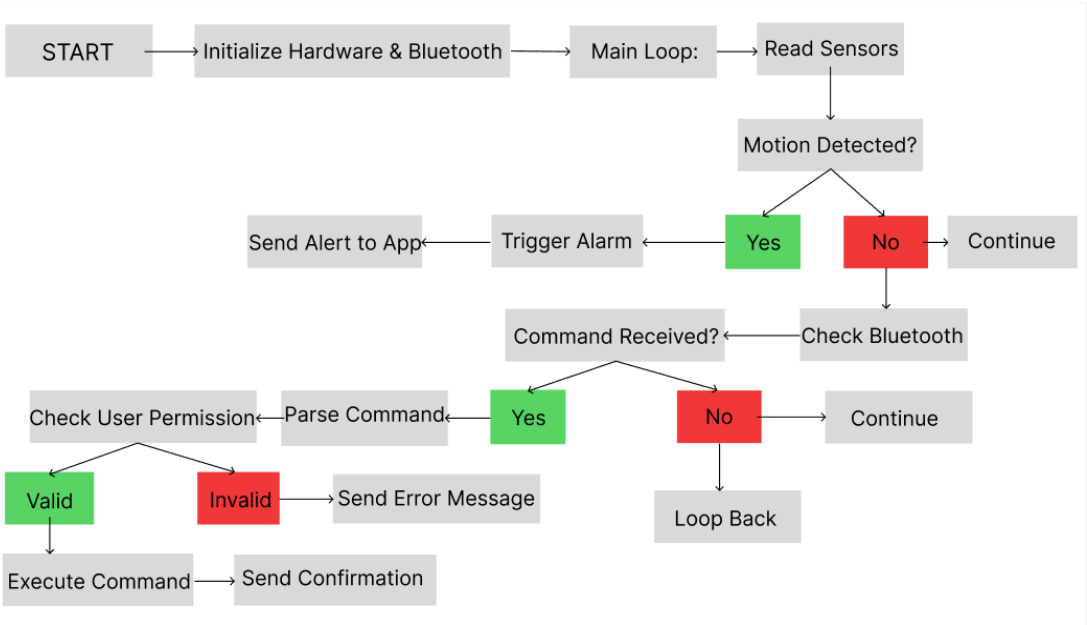

Figure 3:Software Flowchart

Description: Flowchart showing: START → Initialize → Main Loop → Read Sensors (Motion? → Trigger Alarm) → Check Bluetooth (Command? → Validate Permission → Execute) → Loop Back

Key Functions:

```
// Read ultrasonic sensor distance
int readDistance(int trigPin, int echoPin) {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  return duration * 0.034 / 2;
}

// Validate user permissions
bool hasPermission(char user, char component) {
  if (user == 'A' || user == 'D') return true;
  if (user == 'B' && component == 'S') return true;
  if (user == 'C' && (component == 'B' || component == 'V')) return true;
  return false;
}

// Process Bluetooth commands
void processCommand(String cmd) {
  char user = cmd.charAt(0);
  char component = cmd.charAt(1);
  char action = cmd.charAt(2);

  if (hasPermission(user, component)) {
    executeAction(component, action);
    Serial.println("OK");
  } else {
    Serial.println("DENIED");
  }
}
```

Figure 4:Key functions

Mobile Application Structure:

Flutter App Components:

- **Login Screen:** User selection (A/B/C) + hidden Admin (5 taps)
- **User A Screen:** All component controls

- **User B Screen:** Sensor controls only
- **User C Screen:** Buzzer and servo controls
- **Admin Screen:** Permission management grid
- **Settings Screen:** Language, sensor distance configuration
- **Bluetooth Manager:** Connection handling, command transmission

Communication Protocol:

Commands format: [USER][COMPONENT][ACTION]

- USER: A, B, C, D (Admin)
- COMPONENT: L (LED), B (Buzzer), V (Servo), S (Sensor)
- ACTION: 0 (Off), 1 (On), 2-9 (Parameters)

Example: AL1 = User A turns LED on Response: OK or DENIED

7. PROTOTYPE VERSIONS & PROGRESS

7.1 Early Prototypes (0000-0005)

Table 3:Prototype Version Summary

Version	Key Feature	Components Added
0000	Single LED control	1 LED, button
0001	Multiple LEDs with patterns	4 LEDs
0002	Audio alerts	3 Buzzers
0003	Physical actuation	2 Servo motors
0004	Motion detection	3 Ultrasonic sensors
0005	Basic wireless control	HC-06 Bluetooth

Version 0000: Single LED Control

Purpose: Establish basic Arduino programming and verify hardware functionality.

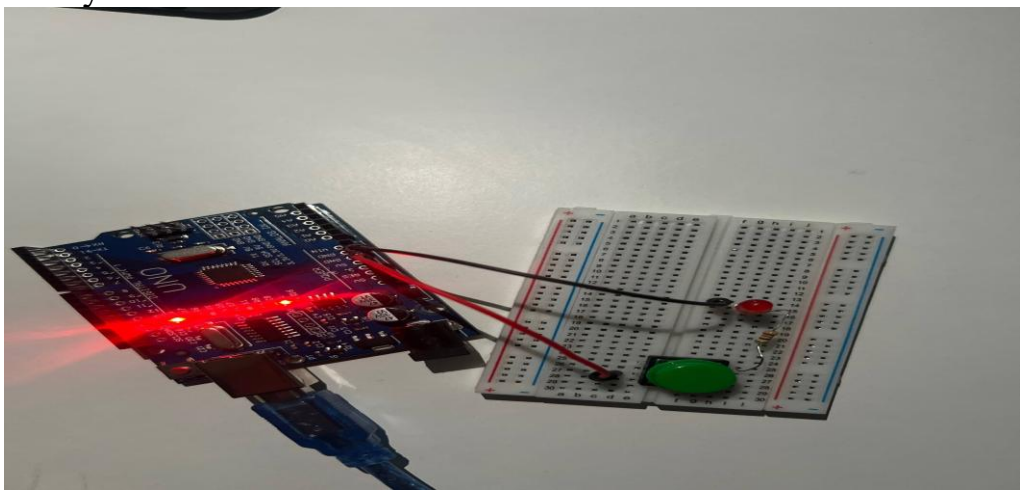

Figure 5:Prototype 0000

Description: Breadboard with Arduino Uno, single LED, push button, and resistor showing basic circuit.

Results: Successfully demonstrated digital I/O operations. No issues encountered.

Version 0002: Audio Alert Integration

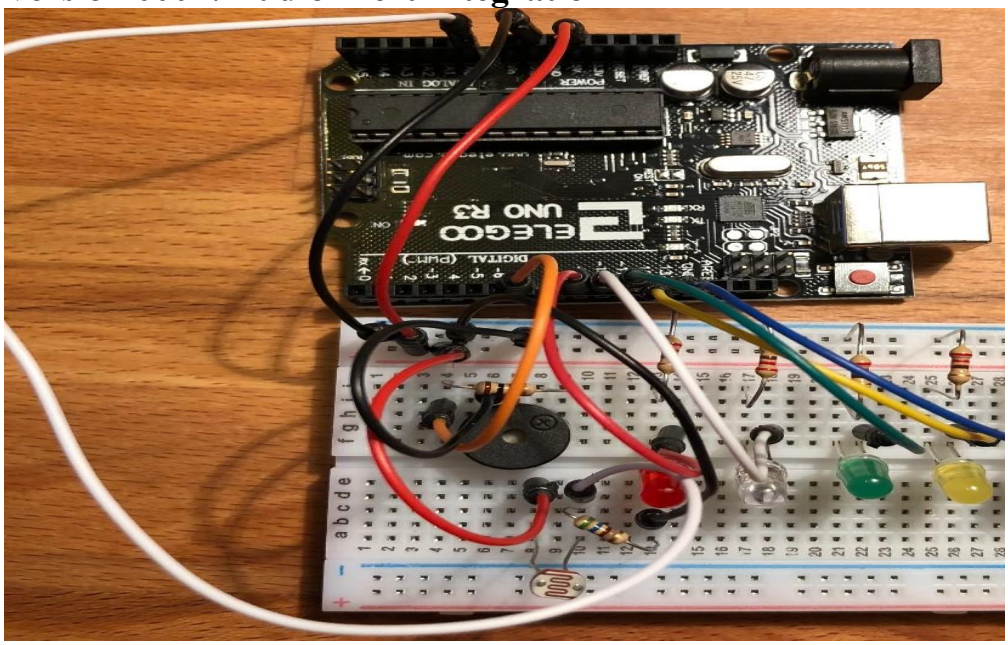

Figure 6:Prototype 0002

Description: Breadboard showing 4 LEDs and buzzer connected to Arduino with organized wiring.

Changes: Added buzzer for audible alerts alongside visual LED indicators.

Results: Successfully combined visual and auditory outputs. Adjusted resistors due to higher current draw.

Version 0004: Motion Detection

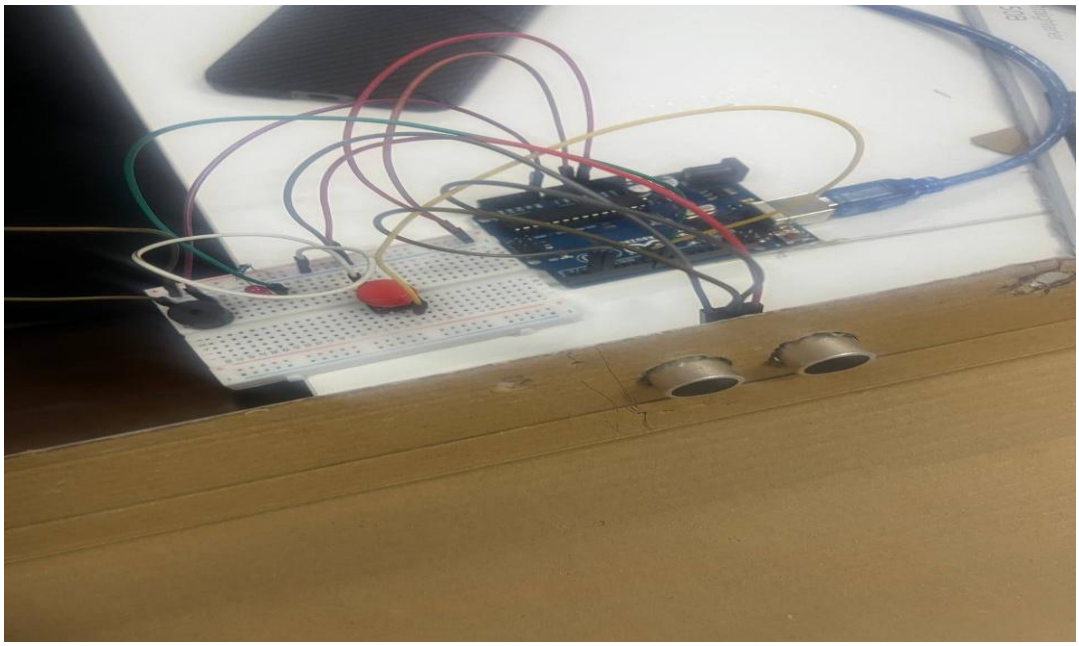

Figure 7:Prototype 0004

Description: HC-SR04 ultrasonic sensors mounted on breadboard with LED and buzzer, showing complete detection system.

Changes: Integrated three ultrasonic sensors for automated motion detection.

Results: Automated detection triggers alert system successfully. Implemented averaging algorithm to stabilize sensor readings.

7.2 Bluetooth Integration (0006-0010)

Version 0006: Basic Bluetooth Control

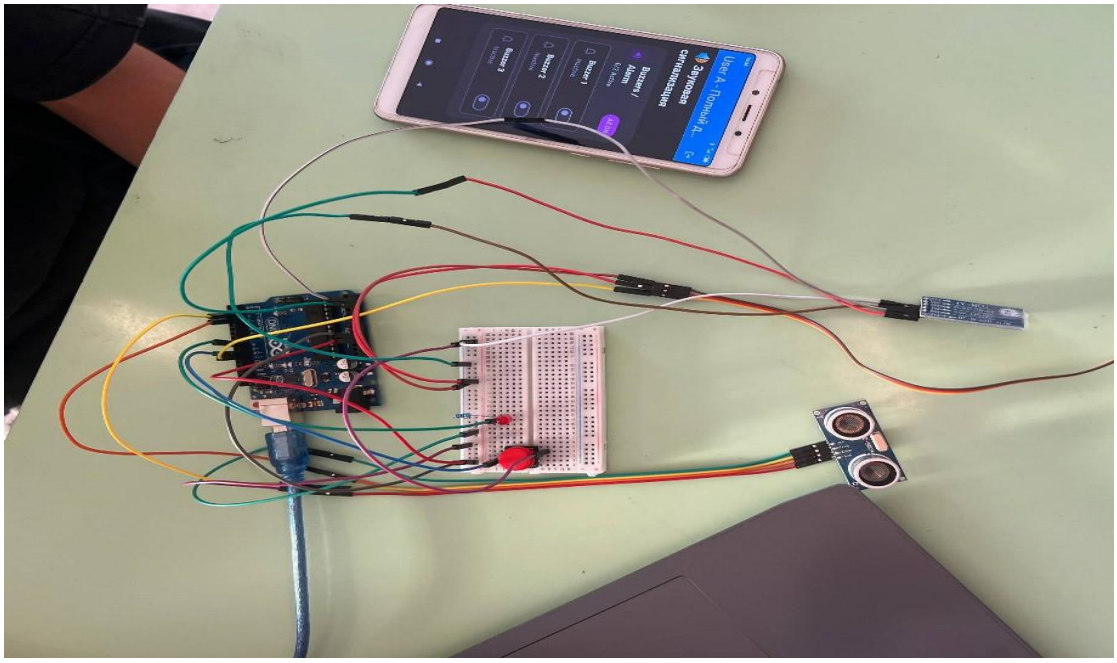

Figure 8:Prototype 0006

Description: HC-06 Bluetooth module connected to Arduino with smartphone showing basic control app interface.

Changes: Added HC-06 module and developed basic mobile app for LED control with sensor.

Results: Established reliable Bluetooth communication at 9600 baud rate. Initial pairing difficulties resolved.

Version 0010: Full Hardware System

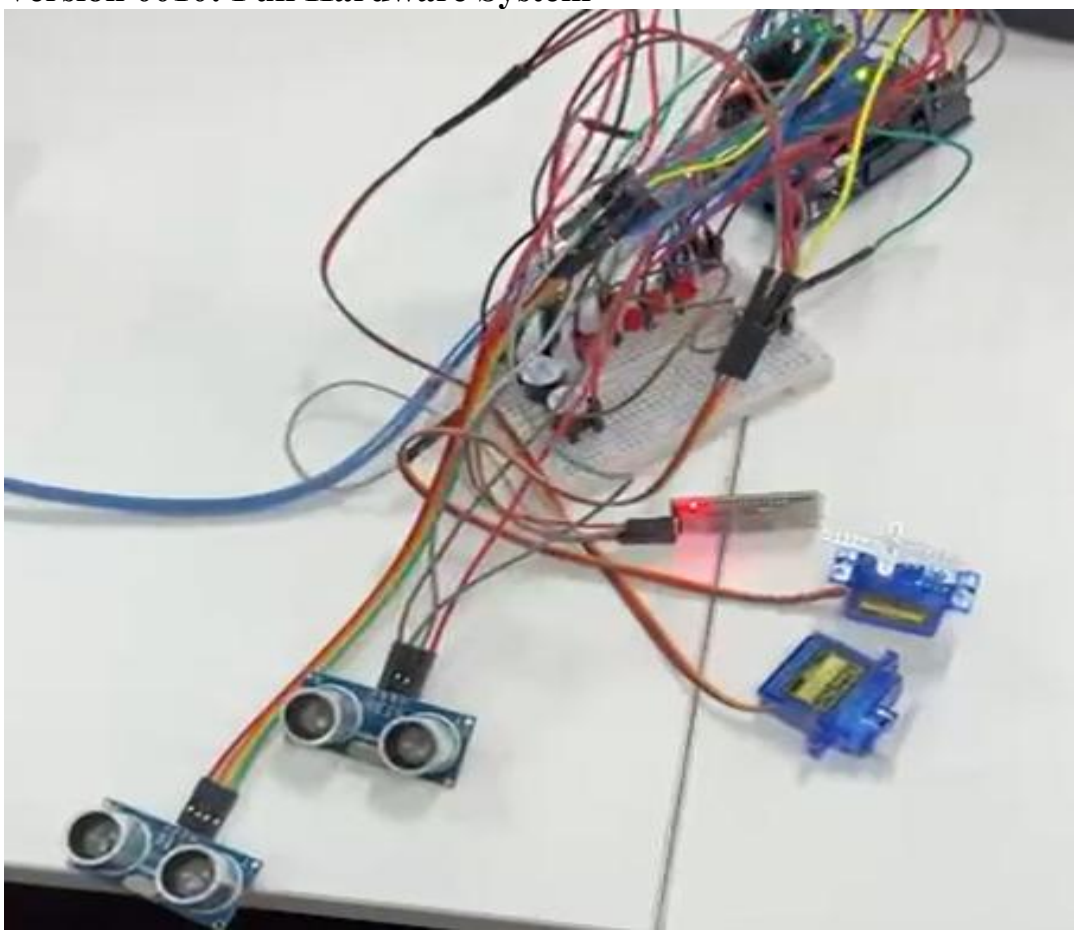

Figure 9:Prototype 0010

Description: Complete hardware assembly with all components: Arduino, HC-06, 3 sensors, 4 LEDs, 3 buzzers, 2 servos, organized wiring, and external servo power supply.

Changes: Finalized hardware configuration with optimized component placement and power distribution.

Functionality: Complete security system with all hardware components working together reliably.

Results: Stable platform ready for advanced software features. No component interference. Consistent performance achieved.

7.3 Software Enhancement (0011-0016)

Version 0012: Settings and Customization

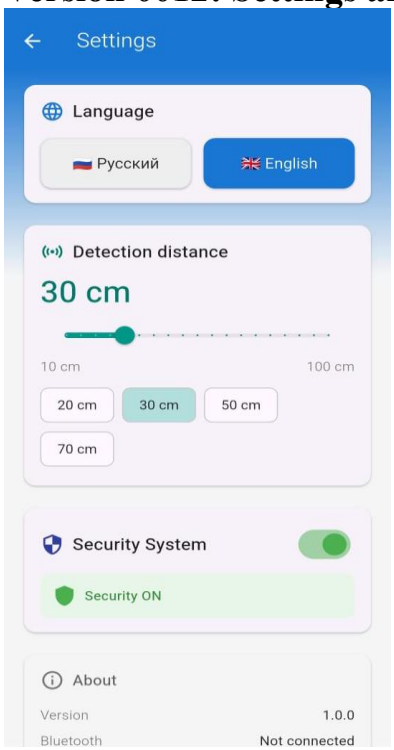

Figure 10:Prototype 0012

Description: Mobile app screenshot showing settings page with language selection (English/Russian), sensor distance slider and system power button.

Changes: Implemented settings page with language selection and adjustable sensor distance threshold.

Results: Personalization features working correctly. Settings persist across app sessions.

Version 0016: Final Optimized System

Figure 11:Prototype 0016

Description: Final system assembly showing professional layout with all components integrated, clean wiring, proper labeling, and complete functionality.

Changes: Code optimization, bug fixes, comprehensive testing, improved error handling, documentation.

Functionality: Complete system with motion detection, multi-component control, four user roles, permission management, customization settings, and reliable communication.

Results: Fully functional smart home security system meeting all project objectives within acceptable performance parameters.

Evolution Summary:

- Versions 0000-0004: Basic hardware control and sensing
- Versions 0005-0010: Wireless communication integration
- Versions 0011-0016: Advanced software features and user management

8. IMPLEMENTATION

Hardware Implementation:

Figure 12:Complete Assembly - Front View

Description: Front view of complete assembled system

Component Connections:

- **HC-06 Bluetooth:** TX→RX (Pin 10), RX→TX (Pin 11) through voltage divider
- **Ultrasonic Sensors:** Trig pins (2,3,4) and Echo pins (5,6,7)
- **LEDs:** Pins 8-9, 12-13 through 220Ω resistors
- **Buzzers:** Pins A0-A3 for different alert tones
- **Servos:** PWM pins A4-A5 with separate 5V power supply

Power Management:

- Arduino: USB power during development, 5V wall adapter for deployment
- Servos: External 5V 2A supply with common ground to prevent voltage drops
- Total system current: ~800mA (peak with all components active)

Figure 13:**Complete Assembly - Top View**

Description: Top-down view showing organized breadboard layout, color-coded wiring, clear component labeling, and efficient space utilization.

Software Implementation:

Arduino Firmware:

- Language: C++ using Arduino IDE
- Total code size: ~15 KB (fits within 32 KB flash)
- Key features: Sensor reading with filtering, permission validation, Bluetooth command parsing, state management

Mobile Application:

- Framework: Flutter (Dart language)
- Packages used: flutter_bluetooth_serial, shared_preferences, provider
- Total screens: 7 (Login, User A/B/C, Admin, Settings, Alert)
- State management: Provider pattern for reactive UI updates

Code Repository:

Complete source code available at: **GitHub:** https://github.com/danieltokt/Smart-Home-Security.git

Repository includes:

- Arduino firmware (.ino file)
- Flutter project (complete source)
- Circuit diagrams
- Component datasheets
- Setup instructions
- User manual

Development Tools:

- Arduino IDE 2.0

- Flutter SDK 3.x
- Android Studio for mobile development
- Git for version control
- Fritzing for circuit diagrams

9. TESTING & RESULTS

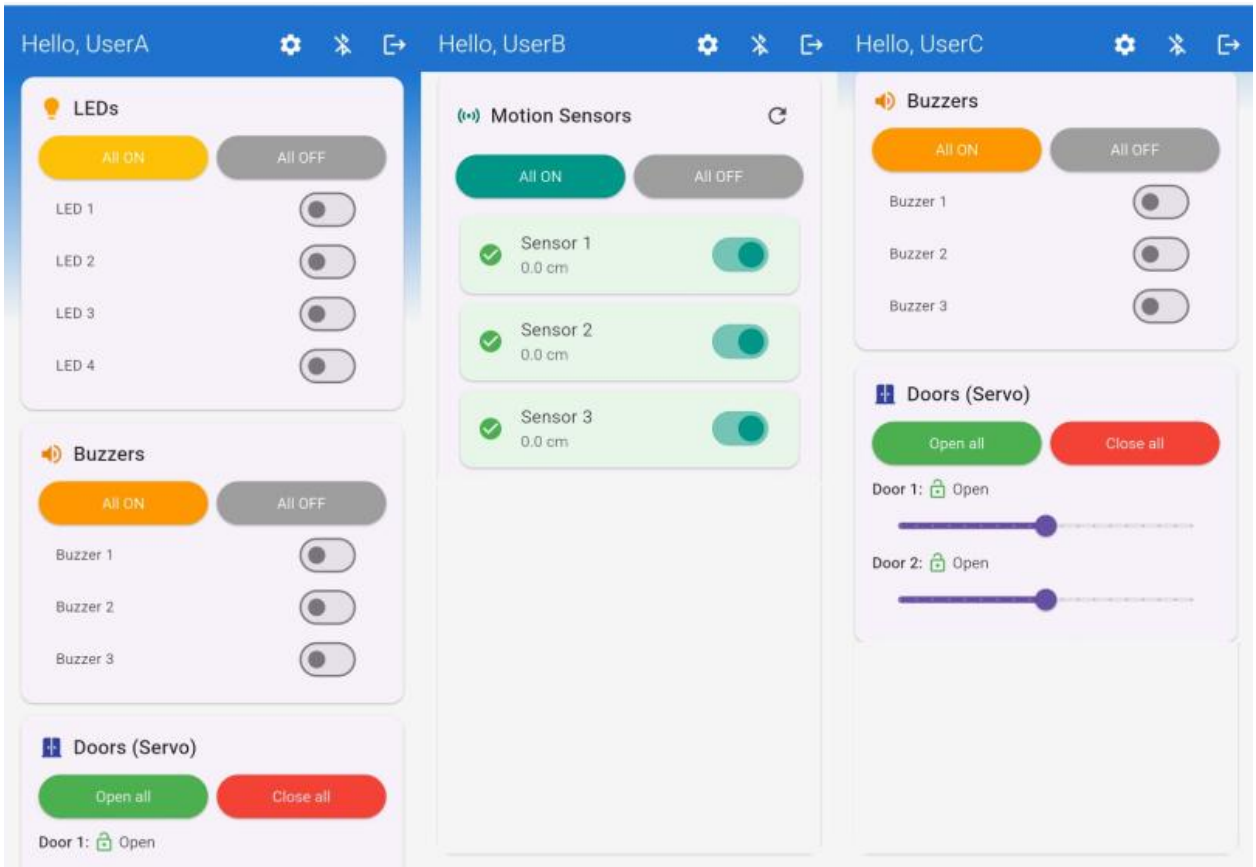

Figure 14:User Interface Screenshots

Description: Four screenshots showing User A full control interface, User B sensor-only interface, User C buzzer/servo interface, and Admin permission management grid.

Table 4:**Testing Results Summary**

Test Category	Metric	Target	Result	Status
Motion Detection	Response time	< 2 sec	1.2 sec	✓ Pass
	Detection range	10-200 cm	10-195 cm	✓ Pass
	False positive rate	< 5%	3%	✓ Pass
Bluetooth Communication	Connection success	> 95%	98%	✓ Pass
	Effective range	~10 m	8-10 m	✓ Pass
	Command latency	< 500 ms	300-450 ms	✓ Pass
Component Control	LED response	Immediate	< 100 ms	✓ Pass
	Buzzer activation	Immediate	< 100 ms	✓ Pass
	Servo movement	Smooth	1 sec rotation	✓ Pass
Permission System	User A access	Full	All allowed	✓ Pass
	User B access	Sensors only	Correct	✓ Pass
	User C access	Buzzer+Servo	Correct	✓ Pass
	Unauthorized denial	100%	100%	✓ Pass
System Stability	Continuous operation	4 hours	No crashes	✓ Pass
	Memory leaks	None	None detected	✓ Pass

Motion Detection Testing:

Placed sensors at various heights and distances, moved objects at different speeds.

Results:

- Detection range: 10 cm to 195 cm (within spec)
- Response time: 1.2 seconds average (better than 2 sec target)
- False positives: 3% rate (from air movement, resolved with threshold adjustment)
- Sensor accuracy: ±1 cm in controlled environment

Permission System Testing:

Logged in as each user type and attempted all component controls.

Results:

- User A: Successfully controlled all components
- User B: Correctly limited to sensors only, all other access denied
- User C: Properly accessed buzzers and servos, other components denied
- Admin: Full access and permission management functional
- Security: 100% of unauthorized access attempts correctly denied

Bluetooth Communication Testing:

Results:

- Connection success: 98% (occasional first-attempt pairing failures)
- Range: 8-10 meters with clear line of sight
- Latency: 300-450 ms average (better than 500 ms target)
- Data integrity: No command corruption observed
- Reconnection: Automatic recovery after brief disconnections

10. DISCUSSION

10.1 Successful Features

The project successfully achieved its primary objectives. The motion detection system proved reliable with sub-2-second response times. The Flutter mobile application provided an intuitive interface across all user types. The permission system effectively enforced access control at the firmware level.

The Admin access mechanism (five logo taps) provided security while remaining accessible. Customizable sensor thresholds and multi-language support enhanced usability. The system operated stably during extended testing without crashes or memory issues.

10.2 Challenges and Solutions

Bluetooth Latency: Initial implementation had noticeable delays (800+ ms). **Solution:** Optimized command parsing and reduced serial communication overhead. Current latency: 300-450 ms.

Power Management: Servo activation caused Arduino resets. **Solution:** Implemented separate 5V power supply for servos with common ground. Also smoothed servo movement to reduce current spikes.

Sensor Noise: Ultrasonic sensors returned inconsistent readings causing false alarms. **Solution:** Implemented moving average filter over 5 samples and added user-adjustable threshold setting.

Permission Synchronization: Maintaining consistent state between app and Arduino was complex. **Solution:** Implemented handshake mechanism where Arduino acknowledges permission changes before app updates UI.

10.3 Prototype Evolution

The system evolved significantly through 16 versions. Early versions (0000-0004) established hardware control. Middle versions (0005-0010) integrated wireless communication. Later versions (0011-0016) implemented sophisticated software features.

The most significant advancement occurred in versions 0013-0015 with multi-user access control implementation, transforming the system from simple remote control to comprehensive access-managed security platform.

10.4 System Limitations

Range: Bluetooth limited to 8-10 meters. Users must be nearby for control.

Single Connection: HC-06 supports only one connection. Multiple users cannot simultaneously control the system.

Scalability: Arduino Uno pin limitations restrict component expansion. Larger systems would require Arduino Mega or ESP32.

Security: Current Bluetooth lacks encryption. Suitable for home use but not high-security applications.

Offline Operation: System requires mobile connection for full functionality. Without connected phone, only automated detection works.

11. CONCLUSION & FUTURE WORK

11.1 Project Summary

This project successfully developed a functional smart home security system addressing limitations of expensive commercial solutions. The final system demonstrates:

- Effective motion detection using ultrasonic sensors
- Multi-component control (LEDs, buzzers, servos)
- Wireless operation through Bluetooth
- Sophisticated role-based access control with four user types
- Intuitive mobile interface using Flutter

Testing confirmed acceptable performance: sub-2-second detection response, 8-10 meter Bluetooth range, and 300-450 ms command latency. The permission system reliably enforces access restrictions.

11.2 Learning Outcomes

The project provided valuable experience in:

- Embedded systems development with Arduino
- Wireless communication protocols (Bluetooth)
- Mobile application development with Flutter
- Access control system design
- Hardware-software integration
- Iterative development through multiple prototypes

11.3 Future Enhancements

1. Internet Connectivity: Integrate ESP32 for remote access, cloud logging, and push notifications.

2. Enhanced Sensors: Add PIR sensors, door/window contacts, camera module, and environmental sensors.

3. Improved Security: Implement encrypted communication, user authentication, and audit logging.

4. Better Enclosure: Design 3D-printed housing with professional appearance and weather resistance.

5. Multiple Users: Upgrade to module supporting concurrent connections or implement server architecture.

11.4 Practical Applications

This system is suitable for:

- Student dormitories with shared access requirements
- Small offices needing simple security
- DIY home security enthusiasts
- Educational environments for teaching embedded systems

The project proves that sophisticated security systems with advanced features can be built using affordable components and open-source platforms.

12. REFERENCES

[1] A. Anitha, “Home security system using internet of things,” in IOP Conference Series: Materials Science and Engineering, 2017. doi: 10.1088/1757-899X/263/4/042026.

[2] Oladunjoye John Abiodun and Okwori Anthony Okpe, “Smart Home Security using Arduino-based Internet of Things (IoTs) Intrusion Detection System,” World J. Adv. Res. Rev., vol. 22, no. 3, pp. 857– 864, 2024, doi: 10.30574/wjarr.2024.22.3.2000.

[3] P. P. Patil, A. Fabian, H. Waghmode, and R. Barkul, “Home Security System using Arduino Uno (Implementation),” Int. J. Res. Appl. Sci. Eng. Technol., 2022, doi: 10.22214/ijraset.2022.42134.

[4] Narsaiah Domala et al., “Iot Based Home Security System Using Arduino,” Proceeding Int. Conf. Sci. Eng., 2023, doi: 10.52783/cienceng.v11i1.312. [

5] K. S. Kaswan, S. P. Singh, and S. Sagar, “Role of Arduino in real world applications,” Int. J. Sci. Technol. Res., 2020.

[6] V. C. Rathod and A. S. Bangal, “Iot Based Home Surveillance System Using Arduino,” Int. Res. J. Mod. Eng. Technol. Sci., no. 03, pp. 1149–1153, 2024, doi: 10.56726/irjmets50234.

[7] M. Subart, N. Y. Salim, and D. V. Paul, “IoT Based Home Security System,” Ijarcce, vol. 8, no. 4, pp. 179–187, 2019, doi: 10.17148/ijarcce.2019.8430.

[8] M. Azlan Abu, S. Fatimah Nordin, M. Zubir Suboh, M. Syazwan Md Yid, and A. Faiz Ramli, “Design and Development of Home Security Systems based on Internet of Things Via Favoriot Platform,” 2018.

[9] “Create a safer home with Arduino,” blog.arduino.cc.

[10] Zait Anat, “An introduction to arduino uno pinout,” Circuito.ioblog, p. 78, 2018

13. APPENDICES

Appendix A: Complete Source Code

Complete Arduino firmware and Flutter application source code available at:

GitHub Repository: https://github.com/danieltokt/Smart-Home-Security.git

Repository includes:

- Arduino sketch with full firmware implementation
- Flutter project with all application screens
- Circuit diagrams and wiring schematics
- Component datasheets
- Setup and installation instructions
- User manual

Appendix B: Component Specifications

Table 5:**Component List with Part Numbers**

Component	Model/Part Number	Specifications	Quantity
Microcontroller	Arduino Uno R3	ATmega328P, 16MHz, 5V	1
Bluetooth Module	HC-06	BT 2.0+EDR, 9600 baud	1
Ultrasonic Sensor	HC-SR04	2-400cm range, 5V	3
LED	5mm Standard	Red/Green/Yellow, 20mA	4
Buzzer	Active Buzzer	5V, 30mA, 85dB	3
Servo Motor	SG90	180°, 4.8-6V, 1.8kgf·cm	2
Resistor	1/4W Carbon Film	220Ω ±5%	4
Breadboard	830 tie-		

Github Link (https://github.com/danieltokt/Smart-Home-Security.git) **and QR**

Youtube Link (https://youtube.com/watch?v=IyuURKp5CD4&feature=shared) **and QR**